2008-10-18

Using Vagueness Measures to Re-rank Documents Retrieved by a Fuzzy Set Information Retrieval Model

Stephen Lynn
stephen.lynn@byu.net

Yiu-Kai D. Ng
ng@cs.byu.edu

Original Publication Citation

Stephen Lynn and Yiu-Kai Ng. "Using Vagueness Measures to Re-rank Documents Retrieved by a Fuzzy Set Information Retrieval Model." In Proceedings of the 5th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'8), Vol. 5, pp. 39-43, October 18-2, 28, Jinan, China.

Using Vagueness Measures to Re-rank Documents Retrieved by a Fuzzy Set Information Retrieval Model

Stephen Lynn
Computer Science Department
Brigham Young University
Provo, Utah 84602, U.S.A

Yiu-Kai Ng
Computer Science Department
Brigham Young University
Provo, Utah 84602, U.S.A

Abstract

Traditional information retrieval (IR) systems evaluate user queries and retrieve/rank documents based on matching keywords in user queries with words in documents. These exact word-matching and ranking approaches ignore too many relevant documents that do not contain the exact keywords as specified in a user query. Instead of considering these traditional approaches, we propose to retrieve documents using a fuzzy set IR model and rank retrieved documents for any vague query using the "vagueness score" of the documents based on the word senses as defined in WordNet. Using the vagueness scores, we rank the most highest "relevant" documents of a vague query q as the ones that best cover the different possible senses of keywords in q. The proposed word-sense ranking method enhances the existing ranking approaches on ordering retrieved documents for vague queries and thus provides a more reliable and elegant tool for information retrieval.

1 Introduction

The main goal of an Information Retrieval (IR) system is to retrieve a set of documents that most closely matches a user's information need. Most traditional IR systems have attempted to achieve this goal by retrieving the set of documents deemed most relevant to a user query according to the appearance of keywords in user queries as in documents [2]. In addition, many IR approaches have been developed to quantify "relevancy," such as using the vector space model [7], so that a system can appropriately rank a set of documents based on matched keywords. This ranking task, however, is incapable of handling documents retrieved by a "vague" query, which imposes multiple interpretations of its intended information need. Instead of considering the traditional ranking approach, we propose to rank documents retrieved for a vague query using the "vagueness score" of the documents, which enhances the existing ranking methods and propagates the most relevant documents of a query high in the ranking. Hence, we shift the traditional IR ranking from ranking the most highest "relevant" documents of a vague query q based on keyword matching to ranking the documents that best cover the different possible senses [3] of the keywords in q. In practice, vague queries have very few keywords and so the total possible query senses to be considered in this approach is manageable in size.

The usage of word senses in ranking a vague query is essential, since in any language, a single word or short phrase is often not enough to determine what it really means. For example, the word "program" can mean "TV program," "concert program," or "computer program." This is because many words have more than one *meaning* (i.e., *sense*), and *word senses* can be found in any dictionary. In fact, possible meanings of a particular word might be closely related or they could be completely different. For example, according to the WordNet lexical database for the English language [6], there are 51 different potential senses for the word 'play.' Anything from a theatrical performance to what a child does with a toy can be represented using this word. Based on this word alone it is impossible to know how the term is being used. This poses a challenging problem for an IR system. For example, given only the word 'play,' it is likewise impossible for an IR system to know the user's real information need. Typical IR systems tend to rank (i) *highest* the documents containing the *most* occurrences of query keywords and (ii) *lowest* the documents including the *least* occurrences of query keywords. For example, if there are more documents with content exclusively on *performance plays* than *playing games*, then a search on the term 'play' returns a list where the top 10 retrieved documents are highly likely about theatrical performances, and the user who looks for information about theatrical plays is satisfied with the retrieved results. On the other hand, the user who looks for information about playing games would likely dissatisfy with the retrieved results. A retrieved document set that is ranked according to the possible number

978-0-7695-3305-6/08 $25.00 © 2008 IEEE
DOI 10.1109/FSKD.2008.546

of unique query-word senses each document covers, however, would not cause the bias as the result of simply using the query-keyword matching to provide the ranking of retrieved documents for a vague query. We develop such a ranking approach based on query senses.

At the heart of the proposed vague-ranking method is the idea of a general *vagueness score*. For sufficiently detailed queries, existing IR approaches are capable of retrieving a relevant set of documents. As explained earlier, the problem arises when a query is *too vague* to accurately determine what the user's information need is. So how can a system decipher when a query is too vague and when it is sufficiently clear? We have developed a sophisticated approach for computing the vagueness score of a user query based on human understandable classifications, which can then trigger a re-ranking mechanism focused on query-sense coverage as opposed to traditional relevance ranking.

We proceed to present our vague-ranking approach as follows. In Section 2, we discuss related work in query vagueness. In Section 3, we introduce our vague-ranking method in re-ranking documents retrieved by an IR model. In Section 4, we present experimental results on our vague-ranking approach. In Section 5, we give a conclusion.

2 Related Work

Alternative approaches to measuring query vagueness have been presented in different work. [4] present an algorithm to calculate query clarity. The proposed model uses a probabilistic approach to compute the entropy between a *query language model* and the corresponding *collection language model*, where the language model is represented by the probability distribution of single-word terms in a document collection. One of the major downsides of this approach is that all of the calculations are based on classifications that cannot be clearly explained or formalized. Hence, none of the *query vagueness concepts* in [4] can be systemically captured to assist a user in clarifying what exactly the user is looking for. However, the intuitive ideas presented in [4] for evaluating the degree of vagueness of a query are helpful in formulating the proposed vague-ranking method.

Another popular method for improving the effectiveness of vague queries is presented in [9], who apply the technique of *query expansion.* In this approach, terms are added to the original query based on either word relationships found in the entire document collection or terms in an initial retrieved set of documents, which yield the global and the local result set expansion strategy, respectively.

Alternatively, [1] discuss another method for query expansion, which is based on *user relevance feedback*. This feedback strategy requires the user to mark documents in the initial retrieved set of a query as either *relevant* or *irrelevant*, and the corresponding IR system then expands the query based on the marked documents. This approach has been shown to be effective in improving query results; however, they tend to favor query senses that are more common in the document collection, even though they might not be relevant to the user's information need. Our query-sense method, on the other hand, does not rely on the user relevance feedback strategy. Instead, we consider the query-sense matching performed between a query and a word-sense dictionary, which can be exploited to outperform the query expansion techniques based on the user's feedback.

3 Our Vague-Ranking Approach

Upon retrieving a set of relevant documents for a user query using our fuzzy set IR model [10], we proceed to re-rank the documents using our vague-ranking approach. Prior to that, it is necessary to establish the key vocabulary to be used in our *vague-ranking* approach and clearly explain how they are used. A fundamental design of our vague-ranking method is *word sense*. Word sense refers to the scenario when a word has more than one meaning or sense based on its context and usage [5]. For example, the word 'tree' can mean anything from a physical plant to the tree data structure used in computer science.

Our vague-ranking approach relies heavily on computing the vagueness scores of query words. The *vagueness score* of a query keyword partially measures the vagueness of the corresponding query based on the number of alternative, possible meanings the keywords in the query might have. A *high* overall score indicates that the corresponding query is *very vague*, whereas a *low* score reflects a *lesser* degree of vagueness of the query. A threshold is used to classify a query as "vague" or "not-vague" based on its vagueness score. This threshold can be system set or exposed to the user based on their perceived information needs.

Besides the vagueness scores, which is formally defined in Section 3.2, a *coverage result set* is computed for each vague query. The coverage result set of a query refers to the same set of documents found in the query result set that is retrieved by an IR system. The difference is that the coverage result set is ordered according to the computed coverage weighting. *Coverage weighting* refers to the weights assigned to documents in a query result set according to the set of *query (word) senses* that the documents cover. Documents that cover query senses and are not considered higher ranked documents in the initial result set receive higher coverage weights. Once a coverage weight has been assigned to each document, the initial result set is re-ranked in descending order according to the coverage weights, and the highest ranked documents in the re-ranked list represent the different possible query senses. The formal definition of the coverage weighting is given in Section 3.3.

3.1 Constructing clusters

In this section we first introduce our word-sense indexing and clustering approach (in Sections 3.1.1 and 3.1.2, respectively). Using the word-sense clusters, we can formally define the *vagueness score* of a user query (in Section 3.2) and the *query-sense coverage* (in Section 3.3).

3.1.1 Word-sense indexing

Prior to computing the vagueness score and determining the coverage weighting for a set of retrieved documents, we first classify the document collection of an IR system into a set of "word-sense" clusters. This step is performed just once on the *entire* document collection, and the resulting clusters of documents based on word senses become available for processing any user query hereafter. The "word-sense" clusters are constructed by first creating an index for each possible word sense in a word-sense dictionary. We have chosen the *WordNet* lexical database, since WordNet is a well-known and widely-used, electronic, word-sense dictionary [6]. Using the set of words and their corresponding word senses found in the WordNet dictionary, we apply the standard TF-IDF vector space model to create a searchable repository of word senses. (The vector-space model is an ideal choice for creating this searchable repository.) Since the proposed vague-ranking approach consists of re-ranking result sets for vague queries, any IR model that returns relevance rankings on retrieved documents for a user query could be used. We choose our fuzzy set IR model [10] due to its fuzzy matching among documents and user queries.

3.1.2 Categorized document in word-sense clusters

After the searchable repository of word senses has been created, each retrieved document d in the collection on which our vague-ranking is applied is examined to determine which *word senses* apply to d. Treating the keywords in a document as a query, we can retrieve a *ranked* list of word senses from the previously created searchable repository of word senses. Using this result set of ranked word senses, our vague-ranking system can assign each document in the collection to the appropriate set of word-sense clusters. Documents with word-sense rankings above a certain threshold value[1] cause each of them in the collection to be added to the associated word-sense cluster such that there is at most one cluster per word sense. The threshold value of 0.95 is chosen so that only the most relevant documents of a word sense are included in their corresponding clusters. Note that a document can be added to more than one cluster as long as words in the corresponding word-sense clusters do not belong to the same grammatical root. For example, a document d can be placed in a cluster of documents related to the word 'tree' as well as to a cluster related to the word 'play'; however, d cannot be placed in two different clusters related to the same word (sense) of 'play.' Without this restriction the system would be unnecessarily dependent on the accuracy of the chosen threshold value. This is because a threshold value that is too *low* would, in the worst case, allow a document d to be added to all of the different word-sense clusters for a given word. During the re-ranking process, d would then appear to cover all of the possible (query) senses and no re-ranking would occur.

This process of assigning different documents to their corresponding word-sense clusters are repeated for each document until all of the documents in the collection have been added to the appropriate word-sense clusters. At this point the system can begin processing user queries.

[1]The ideal threshold value that maximizes the intra-cluster coherency can be found by experimentation.

3.2 Vagueness score

With the word-sense clusters in place, our vague-ranking system can now begin processing user's queries. The first step in this ranking process is in determining the query's vagueness score (*vagueness score* in short). Vagueness scores are computed as follows. Since there is a direct relationship between the number of possible word senses for a query keyword and the vagueness score of the same query, queries that contain words that have *many* different senses should be considered *vaguer* than queries with words that have *few* word senses. Likewise, a user query that has *many* keywords is most likely *less* vague than a query with *fewer* keywords, since more keywords narrow the intended information need of the user. To maintain the balance of these two criterion, the vagueness score is computed as

$$V_q = \frac{1}{|q|} \times \sum_{i=1}^{|q|} \frac{S_i}{S_{max}} \quad (1)$$

where q is a given user query, V_q is the *vagueness score* of q, $|q|$ is the number of the keywords in q, S_i ($1 \leq i \leq |q|$) is the number of *word senses* for the keyword q_i, and S_{max} is the *maximum* number of word senses for any keyword in q.

The *vagueness measure* insures that the impact of S_i on the overall vagueness score is always proportional to the number of keywords in the query. It should also be noted that query keywords that do not exist in the word-sense dictionary (i.e., $S_i = 0$) will not adversely affect the vagueness measure. In fact, the only effect of query keywords without the corresponding word senses is increasing the size of $|q|$, which only reduces the vagueness score of q. This is reasonable, since the likelihood of an *uncommon* keyword that is not found in the word-sense dictionary is *high* and probably indicates that the query should be scored as being *less* vague based on the presence of the query term.

Using Equation 1, the vagueness score of the query q is computed. Queries scoring beneath a certain threshold value, which is 0.65 according to an empirical study, are processed in the standard way, whereas queries with scores above the threshold are classified as vague queries. The result sets corresponding to these vague queries are re-ranked, as described in subsequent sections, so that the final ranking will be based on *query-sense coverage*, instead of the purely document relevancy used by a traditional IR system. We also allow the user to adjust the pre-determined threshold value, i.e., 0.65, based on how sensitive the user prefers our vague-ranking system to be.

Example 1 Consider the keyword query q 'tree pruning.' According to WordNet [6], there are 3 word senses for 'prune' and 7 word senses for 'tree'. Hence, the *vagueness score* of q is $V_q = \frac{1}{2} \times (\frac{3}{7} + \frac{7}{7}) = \frac{5}{7}$. Consider another keyword query q' 'apple tree.' According to WordNet, there is only 1 word sense for 'apple.' Thus, the *vagueness score* of q' is $V_{q'} = \frac{1}{2} \times (\frac{1}{7} + \frac{7}{7}) = \frac{4}{7}$. Based on the vagueness scores of q and q', we conclude that q is more vague than q', which makes sense, since there are more interpretations on the meaning of 'prune' than 'apple.' □

3.3 Query-sense coverage

After a query q is classified as *vague* based on its query-vagueness score, the initial document result set of q, which is retrieved by using our fuzzy set IR model [10], is re-ranked.

3.3.1 Our re-ranking approach

Our re-ranking algorithm begins by iterating through the initial result set of a query and creating a new *coverage result set* based on a coverage weighting. The coverage weighting W_{d_q} of document d with respect to query q is based on (i) the initial relevance score R_d of d computed by using our fuzzy set IR model, (ii) the number of clusters found in the intersection of the *cluster membership*, C_d, of d and the clusters *relevant* to the query, C_q, and (iii) the number of clusters that have already been represented in the set of documents that have been re-ranked, C_u. This *coverage score* ensures that documents that represent previously *unseen* clusters rise to the *top*, whereas documents from clusters that have already been *represented* are *pushed down* in the ranking, even though those documents may have a very high initial relevance score. The *coverage weighting* measure of d with respect to q is defined as follows:

$$W_{d_q} = \alpha R_d + \beta(\frac{|C_d \cap C_q|}{|C_q|}) + \delta(\frac{|(C_d \cap C_q) - C_u|}{|C_u|}) \quad (2)$$

where α, β, and δ are *static weights* such that $\alpha + \beta + \delta = 1$. R_d is the *initial relevance score* of d, C_d is the set of clusters to which d belongs, C_q is the set of clusters relevant to q, and C_u is the set of clusters represented in the current coverage result set. The static weights in Equation 2 provide varying degrees of significance of each component and are independently set by either the end user or experimentation.

The first component in Equation 2 accounts for the *relevance ranking* of the document in the initial result set generated by using our fuzzy set IR model. The initial ranking is included so that documents that are equivalent in their query-sense coverage are ranked appropriately according to their degrees of relevancy. The second component in Equation 2 accounts for how closely related the *word senses* found in d and q are. The *more* query senses they have in common, the *higher* the second component is. The third and final component in Equation 2, which is the crux of the equation, returns a *high* score for d if d contains query senses in q that are previously *unseen* and zero if no previously unseen query senses are found in C_d.

Having a *coverage weighting* in place, the initial result set can be re-ranked. The first document d in the initial result set is added as the first document of the final coverage set, and C_d is added to C_u. For the remaining results in the initial result set, a coverage weighting will be assigned according to Equation 2. After the coverage weight of a document d in the initial result set has been computed, d is added to the re-ranked list in descending order so that documents with a high coverage weight appear first. Eventually, after each of the documents in the initial result set has been evaluated, the re-ranked list is returned to the user.

Example 2 Consider again the keyword query q 'tree pruning' in Example 1 and the initial rankings on a set of documents S as shown in Table 1. Assume that $\alpha = \beta = \delta = \frac{1}{3}$. According to the word-sense clusters of S, C_q = 10 and C_u = 2, and using the initial rankings as shown in Table 1,

$W_{2_q} = \frac{1}{3}(0.995) + \frac{1}{3}(\frac{2}{10}) + \frac{1}{3}(\frac{0}{2})$ = 0.398,
$W_{3_q} = \frac{1}{3}(0.990) + \frac{1}{3}(\frac{2}{10}) + \frac{1}{3}(\frac{0}{2})$ = 0.396, and
$W_{29_q} = \frac{1}{3}(0.855) + \frac{1}{3}(\frac{2}{10}) + \frac{1}{3}(\frac{1}{2})$ = 0.518.

Based on the re-ranked value of each document, the re-ranking of the initial result set is shown in Table 2. □

3.3.2 User presentation

Upon processing a user query, our vague-ranking system provides the definitions for the top N (≥ 1) unique word senses found in C_u, which allow the user an option of selecting one or more of the word senses to re-query the system. The selected word senses can expand the user's initial query and thus redirect the query on results that match the real information need of the user. This feature provides the user an easy way to further clarify the intended meaning in the initial query. Since the top N unique word senses are selected according to the ordering the word-sense clusters in

Doc (d)	R_d	$\frac{\lvert C_d \cap C_q \rvert}{\lvert C_q \rvert}$	$\frac{\lvert (C_d \cap C_q) - C_u \rvert}{\lvert C_q \rvert}$	W_{d_q}
1	-	-	-	-
2	0.995	0.2	0	0.398
3	0.990	0.2	0	0.396
...	...	...	...	...
29	0.855	0.2	0.5	0.518
...	...	...	...	...

Table 1. Initial document rankings and values

Re-Ranked Doc	Comments
Doc_1	Most relevant document is always first. Doc_1 is about pruning plants
Doc_{29}	Doc_{29} is on pruning tree structures
Doc_2	Doc_2 is about the same as Doc_{29}
...	...

Table 2. Re-ranked Document and comments

C_u, the clusters that appear most frequently in C_d for each retrieved document is ordered first. These human readable word-sense definitions are more appealing than interpreting a query word based on a short-document summary in the relevance feedback cycle [1].

4 Experimental Results

The proposed vague-ranking approach was evaluated by running various randomly created test sets of queries through an Internet search engine for gathering a collection of documents and then re-ranking the initial results generated by our fuzzy set IR model to determine how well query-sense coverage is achieved. In the experiment, one of the example keyword queries is "tree pruning", which has been discussed in Examples 1 and 2. Upon processing the query through our fuzzy set IR model on a set of documents retrieved from Google (http://www.Google.com) using the keyword queries 'tree' and 'pruning' independently, a manual re-ranking was performed, which closely matches the re-ranking set generated by using our vague-ranking approach. The relevance ranking values generated by our fuzzy set IR model on the Google documents are partially shown in Table 1 and as discussed in Example 2. Based on the computed coverage weighting scores of the documents retrieved, the results were re-ranked and shown in Table 2.

As shown in Table 2, documents with uncovered senses of query words are bubbled to the top of the re-ranked list. The weights used in the *coverage weighting score* allow the user to determine how specific a query q must be in order for q to be considered *vague*, and the weights demonstrate how important a document that covers new query senses is compared to a document with a high initial relevancy score.

5 Conclusions

Little work has been done in the past on the necessity to achieve the main design goal of an information retrieval (IR) system, i.e., retrieving relevant documents, based on the *vagueness* of a user query. With the average query string length being less than three words [8], the problem of determining the real information need of a user through a query is significant. The proposed *vague-ranking* system in this paper leverages already existing IR systems and provides a new design methodology for those systems, i.e., re-ranking an initial answer set that can assist the user more quickly in locating documents of his real information needs. Our ranking system also facilitates more user's control by allowing the user to adjust the vagueness of weights, and our ranking can favor query-sense coverage or document relevancy in its ranking depending on the user's preference.

References

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

[2] Calado, P., da Silva, A., Vieira, R., Laender, A., Ribeiro-Neto, B.: *Searching Web Databases by Structuring Keyword-Based Queries*. In Proc. of ACM CIKM, pages 26–33, 2002.

[3] Chen, J., Chang, J.: *Topical Clustering of MRD Senses Based on Information Retrieval Techniques*. Association for Comp. Linguistics, 1(24):61–95, 1998.

[4] Cronen-Townsend, S., Zhou, Y., Croft, W.B.: *Predicting Query Performance*. In Proc. of the ACM SIGIR, pages 299–306, 2002.

[5] Fellbaum, C.: *WordNet: An Electronic Lexical Database* (C. Fellbaum, eds.). MIT Press, 1998.

[6] Miller, G.A.: *WordNet Search - 2.1*. Available at http://wordnet.princeton.edu/perl/webwn, 2006.

[7] Raghavan, V., Wong, S.: *A Critical Analysis of Vector Space Model for Information Retrieval*. Journal of the American Society for Info. Sci., 37(5):279-287, 1986.

[8] Sterling, G.: *Search Engine Journal*. Available at http://www.searchenginejournal.com/?p=1563. April 12, 2005.

[9] Xu, J., Croft, W.B.: *Query Expansion Using Local and Global Document Analysis*. In Proc. of the ACM SIGIR, pages 4–11, 1996.

[10] Yerra, R., Ng, Yiu-Kai: *Detecting Similar HTML Documents Using a Fuzzy Set Information Retrieval Approach*. In Proc. of IEEE GrC, pages 693–699, 2005.