Informative Sentence Retrieval for Domain Specific Terminologies

Jia-Ling Koh and Chin-Wei Cho

Department of Information Science and Computer Engineering,
National Taiwan Normal University, Taipei, Taiwan, R.O.C.
jlkoh@csie.ntnu.edu.tw

Abstract. Domain specific terminologies represent important concepts when students study a subject. If the sentences which describe important concepts related to a terminology can be accessed easily, students will understand the semantics represented in the sentences which contain the terminology in depth. In this paper, an effective sentence retrieval system is provided to search informative sentences of a domain-specific terminology from the electrical books. A term weighting model is constructed in the proposed system by using web resources, including Wikipedia and FOLDOC, to measure the degree of a word relative to the query terminology. Then the relevance score of a sentence is estimated by summing the weights of the words in the sentence, which is used to rank the candidate answer sentences. By adopting the proposed method, the obtained answer sentences are not limited to certain sentence patterns. The results of experiment show that the ranked list of answer sentences retrieved by our proposed system have higher NDCG values than the typical IR approach and pattern-matching based approach.

Keywords: sentence retrieval, information retrieval, definitional question answering.

1 Introduction

When students study a course in specific domain, the learning materials usually make mention of many domain specific terminologies. For example, “supervised learning” or “decision tree” are important domain specific terminologies in the field of data mining. The domain specific terminologies represent important concepts in the learning process. If a student didn’t know the implicit meaning of a domain specific terminology, it is difficult for the student to understand the complete semantics or concepts represented in the sentences which contain the terminology. For solving this problem, a student would like to look for some resources to understand the domain specific terminologies. Accordingly, it is very useful to provide an effective retrieval system for searching informative sentences of a domain-specific terminology.

Although various kinds of data on the Internet can be accessed easily by search engines, the quality and correctness of the data are not guaranteed. Books are still the main trustable learning resources of specific-domain knowledge. Furthermore, a book

published in electrical form is a trend in the digital and information age. It makes the electrical books, especially the electrical textbooks, form a good resource for searching the semantic related sentences to a domain-specific terminology in a subject. For this reason, the goal of this paper is to design an effective sentence retrieval system for searching informative sentences of a domain-specific terminology *X* from the given electrical books.

A research topic related to this problem is automatic question answering. The goal of a question answering system is to provide an effective and efficient way for getting answers of a given natural language question. The most common types of queries in English are 5W1H (Who, When, Where, What, Why, and How). For processing different types of queries, various strategies were proposed to get proper answers. The question answering track of TREC 2003 first introduced definitional question answering (definitional QA) [14]. The task of definitional QA system is to answer the questions "What is *X*?" or "Who is *X*?" for a topic term *X*. The problem of searching informative sentences of a domain-specific term is similar to find answers of a definitional question. The typical definitional QA systems commonly used lexical patterns to identify sentences that contain information related to the query target. It has been shown that the lexical and syntactic patterns works well for identifying facet information of query targets such as the capital of a country or the birth day of a person. However, the lexical patterns are usually applicable to general topics or to certain types of entities. As described in [10], an information nugget is a sentence fragment that describes some factual information about the query term. Determined by the type and domain of the query term, an information nugget can include properties of the term, usage or application of the term, or relationship that the term has with related entities. In order to let users understand various aspects of the query term, it is better to cover diverse information nuggets. The pattern matching approach is in direction contrast to discover all interesting nuggets which are particular to a domain-specific term. Therefore, the informative sentences of the term could not be discovered if they didn't match the patterns.

In this paper, for retrieving as complete informative sentences of a domain-specific term as possible, we adopt a relevance-based approach. A term weighting model is constructed in our proposed system by using web resources, including Wikipedia and FOLDOC, to measure the degree of a word relative to the query term. Then the relevance score of a sentence is estimated by summing the weights of the words in the sentence, which is used to rank the candidate answer sentences. By adopting the proposed method, the retrieved answer sentences are not limited to certain sentence patterns. The results of experiment show that the ranked list of answer sentences retrieved by the proposed system have higher NDCG values than the typical IR approach and pattern-matching based approach.

The rest of this paper is organized as follows. The related works are discussed in Section 2. Section 3 introduces the proposed system for retrieving informative sentences of domain-specific terminologies from electrical books. The results of experiment for evaluating the effectiveness of the proposed system are reported in Section 4. Finally, Section 5 concludes this paper and discusses the future work.

2 Related Works

The traditional IR systems provide part of the solution for searching informative data of a specific term, which can only retrieve relevant documents for a query topic but not the relevant sentences. To aim at finding exact answers to natural language questions in a large collection of documents, open domain QA has become one of the most actively investigated topics over the last decade [13].

Among the research issues of QA, many works focused on constructing short answers for relatively limited types of questions, such as factoid questions, from a large document collection [13]. The problem of definitional question answering is a task of finding out conceptual facts or essential events about the question target [14], which is similar to the problem studied in this paper. Contract to the facet questions, a definitional question does not clearly imply an expected answer type but only specifies its question target. Moreover, the answers of definitional questions may consist of small segments of data with various conceptual information called information nuggets. Therefore, the challenge is how to find the information which is essential for the answers to a definitional question.

Most approaches used pattern matching for definition sentence retrieval. Many of the previously proposed systems created patterns manually [12]. To prevent the manually constructed rules from being too rigid, a sequence-mining algorithm was applied in [6] to discover definition-related lexicographic patterns from the Web. According to the discovered patterns, a collection of concept-description pairs is extracted from the document database. The maximal frequent word sequences in the set of extracted descriptions were selected as candidate answers to the given question. Finally, the candidate answers were evaluated according to the frequency of occurrence of their subsequences to determine the most adequate answers. In the joint predication model proposed in [9], not only the correctness of individual answers, but also the correlations of the extracted answers were estimated to get a list of accurate and comprehensive answers. For solving the problem of diversity in patterns, a soft pattern approach was proposed in [4]. However, the pattern matching approaches are usually applicable to general topics or to certain types of entities.

The relevance-based approaches explore another direction of solving definitional question answering. Chen et al. [3] used the bigram and bi-term language model to capture the term dependence, which was used to rerank candidate answers for definitional QA. The answer of a QA system is a smaller segment of data than in a document retrieval task. Therefore, the problems of data sparsity and exact matching become critical when constructing a language model for extracting relevant answers to a query. For solving these problems, after performing terms and n-grams clustering, a class-based language model was constructed in [11] for sentence retrieval. In [7], it was considered that an answer for the definitional question should not only contain the content relevant to the topic of the target, but also have a representative form of the definition style. Therefore, a probabilistic model was proposed to systematically combine the estimations of a sentence on topic language model, definition language model, and general language model to find retrieval essential sentences as answers for the definitional question. Furthermore, external knowledge from web was used in [10] to construct human interest model for extracting both informative and human-interested sentences with respect to the query topic.

From the early 2000s, rather than just made information consumption on the web, more and more users participated in content creation. Accordingly, the social media sites such as web forums, question/answering sites, photo and video sharing communities etc. are increasingly popular. For this reason, how to retrieve contents of social media to support question answering has became an important research topic in text mining. The problems of identifying question-related threads and their potential answers in forum were studied in [5] and [8]. A sequential patterns based classification method was proposed in [5] to detect questions in a forum thread; within the same thread, a graph-based propagation method was provided to detect the corresponding answers. Furthermore, it was shown in [8] that, in addition to the content features, the combination of several non-content features can improve the performance of questions and answers detection. The extracted question-answer pairs in forum can be applied to find potential solutions or suggestions when users ask similar questions. Consequently, the next problem is how to find good answers for a user's question from a question and answer archive. To solve the word mismatch problem when looking for similar questions in a question and answer achieve, the retrieval model proposed in [15] adopted a translation-based language model for the question part. Besides, after combining with the query likelihood language model for the answer part, it achieved further improvement on accuracy of the retrieved results. However, the main problem of the above tasks is that it is difficult to make sure the quality of content in social media [1].

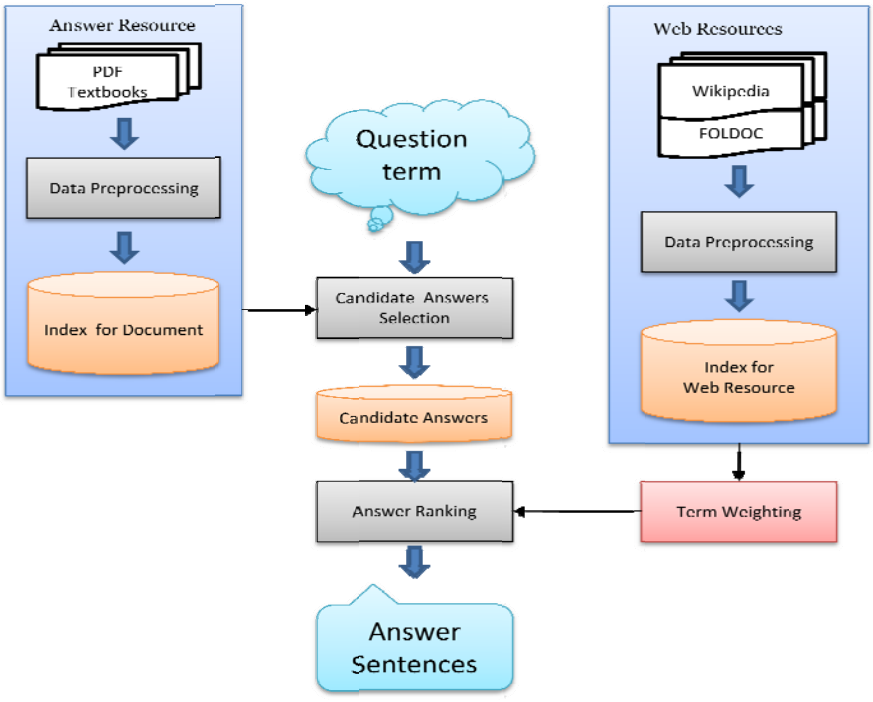

Fig. 1. The proposed system architecture for retrieving informative sentences of a query term.

3 Proposed Methods

3.1 System Architecture

The overall architecture of our proposed system for retrieving informative sentences of a query term is shown as Fig. 1. The main processing components include Data Preprocessing, Candidate Answers Selection, Term Weighting, and Answer Ranking. In the following subsections, we will introduce the strategies used in the processing components in detail.

3.2 Data Preprocessing

<1> Text extraction: In our system, the electrical books in PDF format are selected as the answer resources. Therefore, the text content in the books is extracted and maintained in a text file per page by using a pdf-to-text translator. In addition, a HTML parser is implemented to extract the text content of the documents got from the web resources.

<2> Stemming and stop word removing: In this step, the English words are all transformed to its root form. The Poster's stemming algorithm is applied to do stemming. After that, we use a stop word list to filter out the common words which do not contribute significant semantics.

<3> Sentence separation: Because the goal is to retrieve informative sentences of the query term, the text has to be separated per sentence. We use some heuristics of sentence patterns to separate sentences, such as capital letter at the beginning of a sentence and the punctuation marks: '?', '!', and '.' at the end of a sentence.

<4> Index construction: In order to retrieve candidate sentences efficiently from the textbooks according to the query terminology, we apply the functions supported by Apache Lucene search engine to construct an index file for the text content of the books. In a document database which consists of large amount of documents, the index file not only maintains the ids of documents in which a term appears, but also the appearing locations and frequencies of the term in the documents. In the scenario considered in our system, the text content in the electrical books forms a large document. We apply two different ways to separate the large document into small documents for indexing: one way is indexing by pages and the other one is indexing by sentences.

Let B denote the electrical book used as the answer resource. The set of pages in B is denoted as $B.pages = \{p_1, p_2, p_3, \ldots, p_n\}$, where p_i denotes the document content of page i in B. The set of sentences in B is denoted as $B.sentences = \{s_1, s_2, s_3, \ldots, s_m\}$, where s_i denotes the document content of the *ith* sentence in B. The method of indexing by pages constructs an index file for $B.pages$; the indexing by sentences constructs an index file for $B.sentences$.

The training corpus consists of the documents got from the web resources: the Wikipedia and the free online dictionary of computing (FOLDOC). We also construct an index file for the training corpus in order to calculate the degree of a word relative to the query terminology efficiently.

3.3 Candidate Answers Selection

Whenever the system gets a query terminology *X*, at first, IndexSearcher method provided by Lucene is used to retrieve candidate sentences according to the constructed index file. As the Boolean model adopted popularly in information retrieval, a naive approach of getting candidate answer sentences is to retrieve the sentences which contain the query term. In addition to the sentences containing the query term, it is possible that the other sentences which are close to the query term in the document represent the related concepts of the query term. For this reason, three approaches are proposed for retrieving candidate answer sentences. The first approach uses the index file which is constructed according to the indexing by sentences. Only the sentences which contain the query term will become candidate sentences. The first retrieval method is denoted as "Sentence" method in the following. The second approach retrieves not only the sentences which contain the query term but also their previous two sentences and their following two sentences to be candidate sentences, which is denoted as "Sentence+-2" method. The last approach, denoted as "Page" method, uses the index file which is constructed according to the indexing by page. By adopting the last approach, the sentences in a page where the query term appears will all become candidate sentences.

3.4 Term Weighting and Answer Ranking

In order to perform ranking for the retrieved candidate sentences, the next step is to estimate the relevance scores of the candidate sentences to the query term. Therefore, we use the web resources, including the Wekipedia and the FOLDOC, as the training corpus for mining the relevance degrees of words with respect to the query term.

The term frequency-inverse document frequency (TF-IDF) of a term is a weight often used in information retrieval to evaluate the importance of a word in a document within a corpus. The Lucene system also applies a TF-IDF based formula to measure the relevance score of the indexing objects, the sentences here, to a query term. Therefore, in our experiments, we use the sort method supported by Lucene as one of the baseline methods for comparison.

In our approach, we consider the words which appear in a sentence as features of the sentence. A sentence should have higher score when it has more words with high ability to distinguish the sentences talking about the query term from the whole corpus. Therefore, we apply the Jensen-Shannon Divergence (JSD) distance measure to perform term weighting, which was described in [2] to extract important terms to represent the documents within the same cluster. The weight of a term (word) *w* with respect to the query terminology *X* is estimated by measuring the contribution of *w* to the Jensen-Shannon Divergence (JSD) between the set of documents containing the query term and the whole training corpus.

Let *P* denote the set of query related documents returned by Lucene, which are found out from the training corpus. Let the set of documents in the training corpus be denoted by *Q*. The JSD term weighting of a word *w*, denoted as $W_{JSD}(w)$ is computed according to the following formula:

$$w_{JSD}(w) = \frac{1}{2}\left(p(w|P)\cdot\log\frac{p(w|P)}{p(w|M)} + p(t|Q)\cdot\log\frac{p(w|Q)}{p(w|M)}\right) \qquad (1).$$

The equations for getting the values of $p(w|P)$, $p(w|Q)$ and $p(w|M)$ are defined as the following:

$$p(w|P) = \frac{tf(w,P)}{|P.words|} \quad (2),$$

$$p(w|Q) = \frac{tf(w,Q)}{|Q.words|} \quad (3),$$

$$p(w|M) = \frac{1}{2}\big(p(w|P) + p(w|Q)\big) \quad (4),$$

where $tf(w,P)$ and $tf(w,Q)$denote the frequencies of word *w* appearing in *P* and *Q*, respectively. Besides, $|P.words|$ and $|Q.words|$denote the total word counts in *P* and *Q*, respectively.

According to the JSD term weighting method, the JSD weight of each word in the candidate sentences is evaluated. The relevance score of a candidate sentence *s*, denoted as $Score(s)$, is obtained by summarizing the JSD weights of the words in *s* as the following formula:

$$Score(s) = \sum_{w \in s} w_{JSD}(w) \quad (5),$$

where *w* denote a word in *s*. The top scored sentences are then selected as the informative sentences of the given domain-specific terminology.

4 Experiments and Results

4.1 Experiment Setup and Evaluation Metric

We used the electrical books "Introduction to Data Mining and Knowledge Discovery" and the first four chapters of "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data" as the answer resources. Both books cover the related techniques of data mining, which consist of 222 pages in total.

We implemented three versions of the proposed system according to the three methods of candidate sentences retrieving, which are denoted by “Sentence”, “Sentence+-2”, and “Page”, respectively. The sort method supported in Lucene system was used as a baseline method. Furthermore, a pattern-based approach was also implemented as another baseline.

In the experiment, the following 6 important domain specific terminologies: "Web Mining", "Supervised Learning", "Neural Network", "Naïve Bayesian Classification", "Clustering" and "Decision Tree" were selected to be the test terms. Each test term was used to be a query inputted to the system, where the top 25 sentences in the ranking result provided by our system are returned to be the answers.

We invited 8 testers to participate the experiment for evaluating the quality of the returned answer sentences. All the testers are graduate students in the university, whose majors are computer science. Besides, they are familiar with the field of data mining. For each test term, the sets of the top 25 answer sentences returned by the 5 different methods are grouped together. An interface was developed to collect the satisfying levels of the testers for each returned answer, where the meanings of the 5 satisfying levels are defined as the following.

Level 5: the answer sentence explains the definition of the test term clearly.
Level 4: the answer sentence introduces the concepts relative to the test term.
Level 3: the answer sentence describes the test term and other related terms, but the content is not very helpful for understanding the test term.
Level 2: the content in the answer sentence is not relative to the test term.
Level 1: the semantics represented in the answer sentence is not clear.

4.2 Experimental Results and Discussion

We evaluated the proposed methods by computing Normalized Discounted Cumulative Gain (NDCG) of the ranked answer sentences. The equation for getting NDCG at a particular rank position *n* is defined as the following:

$$NDCG_n = \frac{DCG_n}{IDCG_n} \quad (1).$$

The DCG_n denotes the Discounted Cumulative Gain accumulated at a particular rank position *n*, which is defined as the following equation:

$$DCG_n = rel_1 + \sum_{i=2}^{n} \frac{rel_i}{log_2 i} \quad (2).$$

The rel_i in the equation denotes the actual relevance score of the sentence at rank *i*. We estimated the actual relevance score of a sentence by averaging the satisfying scores given by the 8 testers. In addition, the $IDCG_n$ is an ideal DCG at position *n*, which occurs when the sentences are sorted according to the descending order of their actual relevance scores. In the experiment, the top 25 answer sentences are returned by each method to compute $NDCG_{25}$ for evaluating the quality of the returned sentences. The results of experiment are shown in Tab. 1.

As shown in Tab. 1, it is indicated that the answers retrieved by our proposed method are better than the ones retrieved by either the Lucene system or the pattern-based approach. The main reason is that a sentence usually consists of dozens of words at most. In the Lucene system, a TF-IDF based formula is used to measure the relevance scores of a sentence to the query. Accordingly, only the sentences which contain the query terms are considered. As shown in Fig. 2, the short sentences which contain all the query terms have high ranks in the Lucene system but do not have important information nuggets. On the other hand, most of the sentences retrieved by the JSD term weighting method proposed by this paper describe important concepts relative to the query terminology, which are not limited to specific pattern. In addition, it is not required that the answer sentences must contain the query term. The top 5 answers returned by "Sentence+-2" are shown in Fig. 3 when querying "supervise learning". Furthermore, it is shown in Tab. 1 that none of the three methods for retrieving candidate sentences is superior to the other two methods in all cases.

5 Conclusion and Future Work

In this paper, an effective sentence retrieval system is provided to search informative sentences of a domain-specific terminology from electrical books. A term weighting

Table 1. The $NDCG_{25}$ values of the returned answers of different methods.

	Page	Sentence	Sentence+-2	Pattern	Lucene
"Web Mining"	0.95565	0.94099	0.93864	0.68087	0.75688
"Supervise Learning"	0.87728	0.89939	0.90548	0.73065	0.36578
"Neural Network"	0.95748	0.95172	0.95162	0.67896	0.77962
"Naive Bayesian Classification"	0.85521	0.85960	0.91597	0.76243	0.66769
"Decision Tree"	0.93926	0.92211	0.86469	0.73763	0.64161
"Clustering"	0.85544	0.85809	0.86029	0.63042	0.67952
Average	0.90672	0.90532	0.90611	0.70350	0.64852

1. Supervised Learning classification.
2. Partially Supervised Learning 5.
3. Partially Supervised Learning 5.
4. Partially Supervised Learning 1.
5. Partially Supervised Learning 5.

Fig. 2. The top 5 answers of "Supervised Learning" returned by Lucene.

1. In supervised learning, the learning algorithm uses labeled training examples from every class to generate a classification function.
2. This type of learning has been the focus of the machine learning research and is perhaps also the most widely used learning paradigm in practice.
3. Supervised learning is also called classification or inductive learning in machine learning.
4. Supervised Learning computational learning theory shows that maximizing the margin minimizes the upper bound of classification errors.
5. Bibliographic Notes Supervised learning has been studied extensively by the machine learning community.

Fig. 3. The top 5 answers of "Supervised Learning" returned by "Sentence+-2".

model is constructed in the proposed system by using the web resources, including Wikipedia and FOLDOC, to measure the degree of a word relative to the query terminology. Then the relevance score of a sentence is estimated by summing the weights of the words in the sentence, which is used to rank the candidate answer sentences. By adopting the proposed method, the retrieved answer sentences are not limited to certain sentence patterns. Accordingly, informative sentences of a domain-specific term with various information nuggets can be retrieved as complete as possible. The results of experiment show that the ranked answer sentences retrieved by the proposed system have higher NDCG value than the typical IR approach and pattern-matching based approach.

Among the retrieved informative sentences of a query term, some of the sentences may represent similar concepts. How to cluster the semantics related sentences of a query terminology for providing an instructive summarization of the term is under our investigation currently.

References

1. Agichtein, E., Castillo, C., Donato, D., Gionis, A., Mishne, G.: Finding High-Quality Content in Social Media. In Proc. the international conference on Web Search and Data Mining (WSDM), 2008.
2. Carmel, D., Roitman, H., Zwerdling, N.: Enhancing Cluster Labeling Using Wikipedia. In Proc. the 32nd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR), 2009.
3. Chen, Y., Zhou, M., Wang, S.: Reranking Answers for Definitional QA Using Language Modeling. In Proc. the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, 2006.
4. Chi, H., Kan, M.-Y., Chua, T.-S.: Generic Soft Pattern Models for Definitional Question Answering. In Proc. the 28th international ACM SIGIR conference on Research and development in information retrieval (SIGIR), 2005.
5. Cong, G., Wang, L., Lin, C.Y., Song, Y.I., Sun, Y.: Finding Question-Answer Pairs from Online Forums. In Proc. the 31st international ACM SIGIR conference on Research and development in information retrieval (SIGIR), 2008.
6. Denicia-carral, C., Montes-y-gómez, M., Villaseñor-pineda L., Hernández, R.G.: A Text Mining Approach for Definition Question Answering. In Proc. the 5th International Conference on Natural Language Processing, (FinTal 2006), 2006.
7. Han, K.S., Song Y.I., Rim, H.C.: Probabilistic Model for Definitional Question Answering. In Proc. the 29th international ACM SIGIR conference on Research and development in information retrieval (SIGIR), 2006.
8. Hong, L. Davison, B.D.: A Classification-based Approach to Question Answering in Discussion Boards. In Proc. the 32nd international ACM SIGIR conference on Research and development in information retrieval (SIGIR), 2009.
9. Ko, J., Nyberg E., Si, L.: A Probabilistic Graphical Model for Joint Answer Ranking in Question Answering. In Proc. the 30th international ACM SIGIR conference on Research and development in information retrieval (SIGIR), 2007.
10. Kor, K.W., Chua, T.S.: Interesting Nuggets and Their Impact on Definitional Question Answering. In Proc. the 30th international ACM SIGIR conference on Research and development in information retrieval (SIGIR), 2007.
11. Momtazi, S., Klakow, D.: A Word Clustering Approach for Language Model-based Sentence Retrieval in Question Answering Systems. In Proc. the 18th ACM international conference on Information and knowledge management (CIKM), 2009.
12. Sun, R., Jiang, J., Tan, Y.F., Cui, H., Chua, T.-S., Kan, M.-Y.: Using Syntactic and Semantic Relation Analysis in Question Answering. In Proc. the 14th Text REtrieval Conference (TREC), 2005.
13. Voorhees, E. M.: Overview of the TREC 2001 Question Answering Track. In Proc. the 10th Text REtrieval Conference (TREC), 2001.
14. Voorhees, E. M.: Overview of the TREC 2003 Question Answering Track. In Proc. the 12th Text REtrieval Conference (TREC), 2003.
15. Xue, X., Jeon, J., Croft, W.B.: Retrieval Models for Question and Answer Archives. In Proc. the 31st international ACM SIGIR conference on Research and development in information retrieval (SIGIR), 2008.