A Domain-Specific Concept-Based Searching System

Tru H. Cao, Mai T.H. Ta and Tung Q. Luong
Faculty of Information Technology
Ho Chi Minh City University of Technology
Viet Nam
tru@dit.hcmut.edu.vn

Abstract. Search engines are helpful tools for finding information from a large collection of electronic documents. However, current keyword-based engines, in which searching is basically to match the keywords in queries and those in documents, are not performing well in terms of the precision and recall criteria. The main reason is that, in natural language, one word can have different senses and different words can mean the same thing in particular contexts. This paper is to share our experience in developing a document searching system that aims at answering to "what a user means" rather than "what a user says". Here, queries and documents are abstracted into vectors of pre-defined concepts and, posed a query, the answers will be those documents whose concept vectors and that of the query have the smallest angles. It requires a corpus of specific knowledge about the concepts and their associated terms in a domain of discourse. Our chosen documents are about regulations of financial management in Vietnam. Experiments have shown that the implemented concept-based searching system outperforms traditional keyword-based ones.

1. Introduction

In a computerized society today, electronic documents have become so popular. On one hand, that helps to make paperwork management much more efficient than before, as documents can be stored easily and compactly, and modified and exchanged very fast. On the other hand, that creates a demand to find information from a large collection of electronic documents precisely and sufficiently for a request. Therefore, many text information retrieval models and systems have been proposed and implemented.

However, the majority of current search engines can be classified as keyword-based ones, in which searching is basically to match the keywords in queries and those in documents. They are helpful but are not satisfactory in terms of the precision and recall measures. That is, they miss documents that could be answers to queries, and often return many useless documents that have nothing to do with what a user wants to look for. The main reason is that, in natural language, one word can have different meanings and different words can mean the same thing in particular contexts.

As a natural turn, recently attention has been focused on the semantics-based approach aiming at answering to "what a user means" rather than "what a user says". In [2], conceptual graphs ([8]) were proposed to represent the summarized meanings of documents and queries, and searching was performed as graph matching. Previously, in [9], the notion of concepts was introduced into a searching process, where a concept was defined by a synset, i.e., set of synonymous words, of WordNet ([4]). A document or a query was represented by a vector on a concept space, where the weight for each dimension was determined by the occurrence of the words in the synset defining that concept in the document or the query. Searching was then performed by calculating the cosine of the angle between the document and query vectors.

There are two main shortcomings of the above concept-based method. Firstly, WordNet comprises lexical data with general meanings, while a word can have a specific meaning in a particular domain. Secondly, only terms included in the synset defining a concept are considered to be associated with that concept. Consequently, as presented in [9], the method did not generally outperform the traditional vector model and needed further improvement, especially for a specific domain.

Meanwhile, in [6], concepts were extracted, and the semantic distance of a term to each concept was defined, by experts in a domain of discourse. A drawback of that work is that such a weight assignment to a concept-term association is subjective and hard to be justified. An improvement was carried out by [7], where each paragraph of a document was annotated by a subset of concepts that the paragraph talked about. The weight of a term to a concept was then derived from the occurrence frequency of the term in the paragraphs annotated by the concept. Annotation was however a formidable task to do.

In this paper we present a concept-based searching system for a specific domain, namely, Vietnamese regulations of financial collection and allocation in an organization. The concepts and associated terms in the system are acquired from experts in the field. The Bucket algorithm introduced in [9] is then applied to determine the weight of a term associated with a concept.

The paper is organized as follows. Section 2 reviews traditional keyword-based methods. Section 3 introduces our method in contrast to previous concept-based ones. Section 4 presents and discusses experimental results. For comprehensibility, Vietnamese terms are translated into English. Finally, Section 5 concludes the paper and suggests future work.

2. Keyword-Based Methods

This section summarizes the basic notions of the Boolean and vector models for information retrieval. More details can be found in [1] and [5].

2.1. Boolean Model

One of the simplest keyword-based models is the Boolean one, which is based on set theory and Boolean algebra. Given t as the number of index terms in the domain, a document d is represented by a t-dimensional vector $(w_{1d}, w_{2d},\ldots, w_{td})$, where w_{id}, called the weight associated with the index term k_i, is 1 if k_i is present in d, and is 0 otherwise.

Meanwhile, a query is represented by a Boolean expression in the disjunctive normal form each conjunctive component of which is a t-dimensional vector over the index term set. For example, suppose the index term set has three terms, namely, $\{k_1, k_2, k_3\}$, and a query q is $k_1 \wedge (k_2 \vee \neg k_3)$. Then the disjunctive normal form of q is $(1, 1, 1) \vee (1, 1, 0) \vee (1, 0, 0)$.

The similarity of a document d to a query q, denoted by $sim(d, q)$, is defined to be 1 if the vector representing the document fully matches with the vector representing the query, and 0 otherwise. For example, with the query q as given above and a document d including only the index term k_2, the similarity of d to q is 0.

The advantage of the Boolean model is its clean formalism and simplicity. However, it has two major drawbacks. Firstly, its matching measure is binary, in stead of a grading scale, which is not natural and satisfactory to human thinking. Secondly, it is not easy to translate a document or a query into a Boolean expression.

2.2. Vector Model

The vector model overcomes the first shortcoming of the Boolean model by allowing non-binary index term weights and matching degrees. That is, a document or a query is represented by a vector over an index term set of discourse as in the Boolean model, but the index term weight corresponding to each dimension of the vector is a value in [0, 1].

Let the vector representing a document d be $(w_{1d}, w_{2d},\ldots, w_{td})$ and that representing a query q be $(w_{1q}, w_{2q},\ldots, w_{tq})$. Then the similarity of d to q is defined by the cosine of the angle between these two vectors, that is:

$$sim(d, q) = \frac{\sum_{i=1,t} w_{id} \times w_{iq}}{(\sum_{i=1,t} w_{id}^2 \times \sum_{i=1,t} w_{iq}^2)^{1/2}}$$

Index term weights are derived as follows. Let N be the total number of documents in the system, n_i be the number of documents where the index term k_i occurs, and $freq_{id}$ be k_i's raw frequency, i.e., the number of times k_i occurs in d. The normalized frequency of k_i in d is defined by:

$$tf_{id} = freq_{id} / max_l \{freq_{ld}\}$$

where the maximum is computed over all the terms that occur in d. If the term k_i does not occur in d, then $tf_{id} = 0$.

The inverse document frequency for k_i is defined by:

$$idf_i = log(N / n_i)$$

While tf_{id} quantifies the occurrence degree of k_i in d, idf_i measures the significance of the occurrence of k_i in a document; the more the number of documents where k_i occurs is, the less significant the occurrence of k_i is. So the weight of k_i to d is defined by:

$$w_{id} = tf_{id} \times idf_i$$

For a query q, the weight of k_i to q is suggested to be:

$$w_{iq} = (1 + tf_{iq}) \times idf_i / 2$$

where tf_{iq} is the normalized frequency of k_i in q.

The vector model is popular nowadays for a number of reasons: (1) its partial matching measure allows approximate answers and their ranking; (2) its evaluation of index term weights improves retrieval performance; and (3) it is simple and fast. However, the model still suffers from the common disadvantage of the keyword-based approach, which relies on the occurrence and exact matching of index terms in documents and queries.

3. Concept-Based Methods

In concept-based methods, vectors are defined over a space of concepts in a domain of discourse, and the meanings of terms are taken into account.

3.1. Concepts as Synonym Sets

One of the first attempts to make a turn from the keyword-based approach is [9], where a concept is defined by a synset in WordNet. A document (or a query) can be represented by a vector over a space of such concepts. The weight for each dimension of that vector can be obtained from the weights of the terms that are associated with the respective concept and occur in the document, which are determined by the Bucket algorithm presented below.

Firstly, each concept is assigned a bucket, and an edge is drawn from a term to the concept whose defining synset contains that term, as illustrated in Figure 3.1. For each term encountered in a document, the weight of the bucket of the concept sharing an edge with that term is increased. After the whole document is scanned, the concepts are ranked with respect to the weights of their assigned buckets. Then the concepts that a term represents in the context of the document are those that share edges with that term and have the highest ranks, i.e., greatest weights.

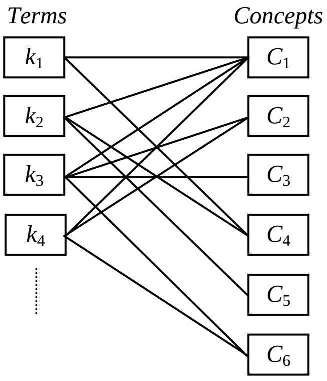

Figure 3.1. Bucket Algorithm Model

On the basis of the ranked concepts associated with each term, the weight associated with each concept of that term can be obtained in a number of ways. One way is the uniform distribution, which assigns the same weight $1/n$ for every concept in the total number of n concepts. As an alternative, the Zipfian distribution assigns the weight $(1/i)/(\sum_{j=1,n}1/j)$ for the i-th ranked concept.

As mentioned in [9], the method does not generally outperform the traditional vector method, especially for a specific domain. In our opinion, that is because: (1) WordNet is not limited to the vocabulary of a specific domain; and (2) the weight of a term to a concept is still determined by the inclusion of the term in the synset defining that concept.

3.2. Concepts as Content Abstractions

In [6] and [7], concepts are abstraction of the contents of documents, which are domain-specific and acquired from experts in the field. In [6], the weight associated with each concept of a term is subjectively assigned. As an improvement, in [7], each paragraph of a document is annotated with some relevant concepts, and the weights of a term to those concepts are determined by its occurrence frequency in the paragraph. However, both of those methods require lots of manual handling that is hard to be justified and carried out.

Here we propose a new way to construct a set of concepts for a specific domain, and to evaluate concept-term weights. Our chosen domain comprises documents about Vietnamese regulations of financial collection and allocation in an organization.

For recognizing relevant concepts in the domain of discourse, we apply the middle-out approach, starting with some easily recognized ones and expanding them with their generalizations and specializations. At the end, only the most specific concepts are retained. This process is illustrated in Figure 3.2, starting with the concept fund. While the concepts in [9] are represented by synsets, our concepts are abstractions of salient points of documents in the system.

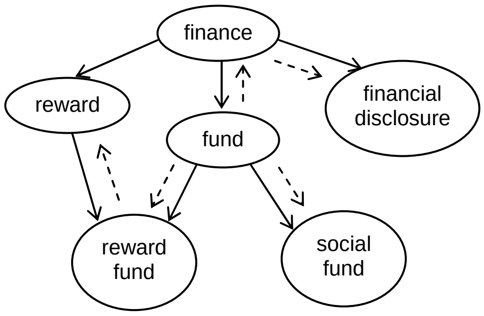

Figure 3.2. Construction of Concepts

Each concept is associated with a set of terms that are related to, or referred by, the meaning of the concept. In order to reduce the term space and allow flexible matching, synonymous terms in the domain of discourse are grouped together and represented by a synset. Those synsets are obtained by analyzing the documents and user's feedbacks. For example, the associated terms of the concept rewarded agents in education include: agent, reward, education, training, individual, collective body, student, lecturer, staff, title, competition.

We employ the Zipfian Bucket algorithm to determine the weight of a term associated with a concept in a document, and retain only the two highest ranked concepts for each term in each document. For a query, as it is too short to make a context, the weight of a term associated with a concept in a query is defined to be the average of the weights of the term to the concept in all the documents of the system. This provides the basis for adapting the *tf.idf* scheme on index terms presented above for concepts as follows.

Let N be the total number of documents in the system, n_i be the number of documents where a concept C_i is retained after Zipfian Bucket weighting. For each document d, let α_{jd} be the weight of the term k_j associated with C_i, and f_{jd} be k_j's raw frequency in d.

We define the raw frequency $freq_{id}$ and the normalized frequency tf_{id} of C_i in d as follows:

$$freq_{id} = \sum_j \alpha_{jd} \times f_{jd}$$

$$tf_{id} = freq_{id} / max_l\{freq_l\}$$

where the sigma is computed over all the terms that share edges with C_i, and the maximum over all the concepts that are retained in d. If, the concept C_i is not retained in d, then $tf_{id} = 0$.

As for an index term in the vector model, the inverse document frequency for C_i is defined by:

$$idf_i = log(N / n_i)$$

and the weight of C_i to d is defined by:

$$w_{id} = tf_{id} \times idf_i$$

from which the concept vector representing d is obtained.

4. Experimental Results

We have realized our concept-based model and compared its performance with the vector model, on the same document and term sets. There are over 142 documents and 500 terms about regulations of finance and budget management in administrative units in Vietnam. Our concept-based system uses about 170 concepts acquired from experts in the field.

The two systems have been tested with 80 queries, compared on the first five returned documents for each query, and based on two criteria, namely, the expected number of returned documents and their ranks. Experiments have shown that they perform the same on 34 queries, the concept-based system performs better on 44 queries and worse on 2 queries. There are 11 queries for which the concept-based system returned expected documents with very high ranks, while none of them was returned by the traditional vector system.

For instance, in order to know what are defined as social organizations, one can pose the query "social organizations". The concept-based system returned the documents that contain actual social organizations with the highest ranks, while the traditional vector system returned only the documents with a high occurrence frequency of the term social organizations. For another instance, with the query "indirect workforce", the traditional vector system returned one document in which the term indirect workforce occurred, while the concept-based system returned also documents about workforce in general, though with lower ranks.

In order to speed up searching, we have employed document clustering techniques and stored the documents in a hierarchical tree ([3]). The precision and recall of our concept-based system, tested with the above-mentioned documents and queries, are respectively 76.79% and 85.99%.

5. Conclusion

We have presented a new concept-based model for document searching that overcomes the drawbacks and combines the advantages of previous models. The concepts and terms in the system are domain-specific and recommended by experts in the field. The cornerstone of our model is that a term is associated with a concept not necessarily because it has a similar sense to the concept, but possibly because it is related to, or referred by, the concept's meaning.

The weights of terms associated with concepts are determined by the Zipfian Bucket algorithm, from which the *tf.idf* scheme for a term space has been adapted for a concept space. The model has also been implemented, demonstrating a better performance than the traditional vector model.

The results reflect the point that intelligence requires knowledge. For running the system on a larger collection of documents in the chosen domain, a corpus with more concepts and terms is needed. Building up such a database of semantic data, in contrast to WordNet database of lexical data, is a project worth investigating for future work.

References

[1] Baeza-Yates, R. & Rebeiro-Neto, B. (1999). *Modern Information Retrieval*. ACM Press, New York.

[2] Cao, T.H. & Nguyen, T.H.D. & Qui, T.C.T. (2004). Searching the web: a semantics-based approach. In Proceedings of the 2003 International Conference on High Performance Scientific Computing, Springer-Verlag, to appear.

[3] Fung, B.C.M. (2002). Hierarchical document clustering using frequent item sets. Master's Thesis, Simon Fraser University.

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

[5] Greengrass, E. (2000). Information retrieval: a survey. Technical Report TR-R52-008-001, CSEE Department, University of Maryland, Baltimore County.

[6] Le, V.T. & Tran, C.N.H (2004). A search engine on regulation documents of financial collection and allocation in an organization. BE Thesis, Ho Chi Minh City University of Technology.

[7] Lu, H.A. (2004). A decision support system for financial collection and allocation in an organization. Master's Thesis, Ho Chi Minh City University of Technology.

[8] J.F. Sowa (1984). *Conceptual Structures - Information Processing in Mind and Machine*. Addison-Wesley Publishing Company.

[9] Whaley, J.M. (1999). An application of word sense disambiguation to information retrieval. Technical Report PCS-TR99-352, Computer Science Department, Dartmouth College.