Document Ranking Method for High Precision Rate

Mee-Sun Jeon and Se-Young Park
Natural Language Processing Section
Electronics and Telecommunications Research Institute
TaeJon, Korea
{msjun,sypark}@com.etri.re.kr

Abstract Many information retrieval(IR) systems retrieve relevant documents based on exact matching of keywords between a query and documents. This method degrades precision rate. In order to solve the problem, we collected semantically related words and assigned semantic relationships used in general thesaurus and a special relationship called keyfact term(FT) manually. In addition to the semantic knowledge, we automatically constructed statistic knowledge based on the concept of mutual information. *Keyfact* is an extended concept of keyword represented by noun and compound noun. Keyfact can be a verb and an adjective including subject or object term. We first retrieved relevant documents with original query using *tf* * *idf* weighting formula and then an expanded query including keyfacts is used in both second document ranking and word sense disambiguating. So we made an improvement in precision rate using keyfact network.

1 Introduction

The general users are more interested in concepts rather than words itself. But many commercial IR systems retrieve relevant documents based on keyword string matching between a query and documents. There are two problems in using the method. The first problem is that words are ambiguous, and this ambiguity is causative of retrieving irrelevant document semantically. Therefore lexical ambiguity has to be resolved. The second problem is that a document is treated as a irrelevant document in spite of a relevant document, for the document does not include the same words as query terms. So an original query has to be expanded to semantically related words.

In order to solve the problems, we consider keyfacts as well as keywords. A keyfact is an extended concept of a keyword and can be defined as a verb and an adjective which are every probability that occurs in several times based on threshold value. We collect semantically related keyfacts from an encyclopedia and assigned semantic relationships using in general thesaurus and a special relationship of FT. We use the semantic informations for document ranking and word sense disambiguation. The remainder of this paper is organized as follows. Section 2 gives a definition of the KN and the KN construction method. Section 3 describes word sense disambiguation. The results of performance comparison are presented in section 4. The concluding remarks are described in section 5.

2 Keyfact Network

2.1 The Definition of KN

In this paper, we collected words and their semantically related words from an encyclopedia(ENCY). Collected informations are used to understand user's request. The KN consists of nodes and edges. Nodes are defined as words rather than word sense and the edges represent binary relationships, such as BT(Broader Term), NT(Narrow Term), RT(Related Term), HP(Has Part), UF(Used For), and FT(keyFact Term). FT is defined as relationship

between noun and verb or adjective. Most IR systems have used to the relationships except FT relationship in their thesaurus. FT is effectively used to resolve lexical ambiguity and compute query-document similarity. KN has a semantic and statistic informations like this format 'relationship(freq(x), freq(y), freq(x,y))'.

2.2. Construction of keyfact network

(ENCY) explains matters systematically. The ENCY has two characteristics. First, it has syntactic characteristic composed of title word and it's explanation part. Second, it has semantic characteristic that most words in the explanation part are semantically related with a title word. ENCY is good to easily collect words and it's semantically related words. We thought that ENCY is a proper text for construction of semantic informations. First, we manually marked keywords like nouns and compound nouns which are semantically related with a title word within each explanation, and we marked keyfacts like verbs and adjectives which are include adjacent subject or object if possible. Second, we assigned semantic relationships using in general thesaurus such as BT, NT, RT, HP, UF and a special relationship of FT. Third, we compute each frequency and co-occur frequency between keyfacts in ENCY. There are 17% ambiguous words of about 22,000 title words. 88% of the ambiguous words are two sense words. KN has 88,010 whole entries, and each entry has an average of 7.3 related words.

For example in the query "팔만 대장경의 자수는? (What's the number of letter in the Tripitaka Koreana) ", the word '자수' might be one of three meanings in Korean language. The three meanings are the number of letter, self-surrender, and embroidery in Korean language. But KN does not have the semantic 'the number of letter', because input text of KN is small corpora. Therefore the KN must be changed and expanded on demand a new ENCY. Whenever a new document like an encyclopedia is occurred, we don't have to process first and second step fully manually. Keyfact extractor extracts noun, compound noun, and original form of verb. If the words like homonym and polysemous word exist in a new document, disambiguator will reduce humane intervention.

3 IR System Based on Keyfact Network

Figure 1 depicts the structure of our IR system. The IR system based on the KN have an advantage. The edges of the KN represent term dependencies more exactly than the conventional statistical measures. So retrieval effectiveness of IR systems can be improved by using KN. Input of our system is natural language query and output is ranked documents.

In our system, we have 5 modules. The system has keyfact extractor, query expander, disambiguator, document ranking subsystem and IR engine. In most other systems nouns(N) and compound nouns(CN) are considered as prime keywords. In this paper, we extract verbs(V) and adjectives(AD) as well as N and CN from queries. Our keyfact extractor extracts N, CN, original form of V and AD. Disambiguator resolves lexical ambiguity in a query. Query expander expands keyfacts extracted from original query into semantically related keyfacts. Expanded queries are consisted with BT, NT, HP, UF, RT, and FT relationship. In IR subsystem we execute retrieve with original query. We use expanded queries in document ranking and applying tf * idf weighting method. Because query expansion is recall enhancing technique, we used expanded query keyfacts when compute query-document similarity. So we got high precision rate.

3.1 Disambiguator

Before query expansion, if a query has polysemy or homonym, their lexical ambiguity have to be resolved. In our disambiguator, we used a knowledge base KN.

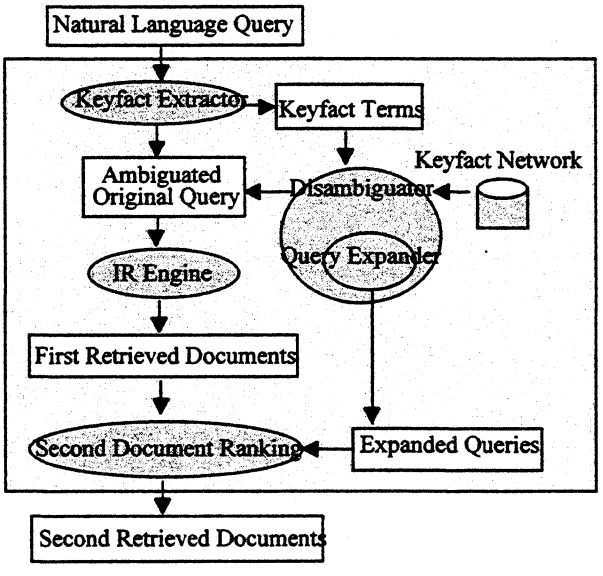
The literature generally divides lexical ambiguity into two types: syntactic and semantic[2]. Syntactic ambiguity refers to differences in syntactic category. Semantic ambiguity refers to differences in meanings. A number of approaches have been taken to word

sense disambiguation. Lesk uses the Oxford Advanced Learners Dictionary[3] and Weiss uses word cooccurrences[4].

In this paper, we resolved semantic ambiguity of N using KN. The ambiguity in a query is resolved when the query is analyzed. Sense resolution is approximated by requiring new terms to be related to at least two original query terms. Shorter queries benefit less than longer queries. Ambiguous words are not able to effectively expand before the ambiguity is not resolved. We resolve ambiguous query terms, and then expand unambiguous query terms. We prepare tables. The number of tables is same as the number of meanings of an ambiguous word registered in KN. If each keyfact lists of each meaning has common keyfact with keyfact lists of the other keywords in a query, the value is simply incremented. A meaning having maximum value over threshold value is chosen as a proper meaning. Let's take the following example sentences.

Example 1) 당나귀와 말의 차이점은 무엇인가?
(What's the difference donkey and horse?)
Example 2) 사람이 마실 수 있는 차의 종류는?
(What kind of tea do man can drink?)

In the example 1, keyfacts extracted from the query are donkey, horse, and difference. Korean language 말 is a polysemous word and has several meanings, which are a language, a horse, the end, a unit of measure, etc. 말's semantic is to be animal according to common keyfact lists in KN.



<Figure 1> The structure of our IR system

Because 말 has four common keyfacts donkey, Equida, tail, and animal, the meaning of 말 is a horse. In this manner keyfacts extracted from the example 2 are man, drink, tea, and a kind. 차's meaning is to a tea according to keyfact drink. In the example 1, keyword concept was useful to disambiguate. In the example 2, keyfact concept was useful.

3.2 Query Expansion

Given a KN, there are a wide choice of words to add to a query vector. One can add only the synonyms, or synonyms plus all descendants, or synonym plus parents and all descendants, or

synonyms plus directly related words, etc. and any number of child links may be traverse. Expansion by synonyms plus and directly related word is benefit[1]. So we choose the parameter. As an example of the expansion processing, let's consider the related words for 'quartz'. If the child's links are limited to depth one, then mineral, rock, be given a pressure, compose rock, like a glass, crystal, and be changed white sands would be added.

3.3 Document Ranking

IR systems must be designed to aid users in determining which documents of those retrieved are most likely to be relevant to given queries. Therefore document ranking is very important part. Most commercial text retrieval systems employ inverted files to improve retrieval speed. The inverted file specifies a document identification number for each document in which the word occurs. In order to improve retrieval effectiveness, vector processing systems employing similarity measures have been suggested and studied extensively. In a vector processing system, An expanded query(EQ) can be represented as vector $<q_1, q_2, \cdots, q_v>$ with original query terms and one depth descendants in KN. The similarity between EQ and documents can be computed in order to rank the retrieved documents in decreasing order of the query-document similarity [5,6]. D_{ij} represents the weights of term *j* in document i. Q_j represents the weights of term *j* in query q. $tf_{Di}(t_j)$ represents the term frequency of term t_j in document Di. $idf(t_j)$ is called the inverse document frequency of term t_j and is set to $\log_2(N/df(t_j))$. N is the number of documents in a collection $df(t_j)$ is the document frequency of term t_j.

$$\text{Similarity}(EQ, D_i) = \sum_{j=1}^{v} D_{ij} * Q_j \qquad (1)$$

$$D_{ij} = tf_{Di}(t_j) * idf(t_j)$$
$$= tf_{Di}(t_j) * \log_2(N/dfj) \qquad (2)$$

Each document vector uses *tf * idf* weighting strategies. W_{ij} in equation 1 is computed by using equation 2 and saved in a inverted file. Thus a term has a high weight in a document if it occurs frequently in the document but infrequently in the rest of collection. The vector processing system allows a query to be expressed as a natural language text describing the user's information need. The description can be treated as a short document so that W_{qj} can be expressed in *tf * idf* weights as well. But to further reduce the computational cost, the weight of original query terms is 0 when a term is absent, and 1 when a term is[7]. The weight of synonym of original query terms is 0.8, and The weight of all descendants of original query terms is 0.3.

In our document ranking subsystem, First we compute original query-document similarity using equation 1. Because D_{ij} is already computed in the inverted file, we retrieve relevant documents by computing only Q_j. Until now we have considered keywords, not considered keyfacts and lexical ambiguity of keywords in a query and documents. Generally expanded queries degrade the precision rate and query expansion is a recall enhancing technique. So we controlled order of already retrieved documents using EQ, which includes keyfacts as well as keywords. For each keyfact in the expanded queries, the system enters the document in a hash table; the table is keyed on the document number, and the value is initially 1. If the documents was previously entered in the table, the value is simply incremented. The end result is that each entry in the table contains total number of keyfacts of expanded queries that occurred in that document. The table is then sorted to produce a ranked list of documents.

4 Performance Comparison

We collected about 600 natural language queries from general users and choose 100 well-formed queries. The rest queries don't have relevant documents in our collection and ill-formed queries. We evaluate the performance of the precision based on KN. The test collection is an encyclopedia and has about 22,000 documents. When we don't use the KN, the precision rate is 66%. when use, the presion rate is 88.8%. Following examples show retrieved documents profited by considering expanded query keyfacts in query-document ranking. Italic style fonts represent relevant documents. In a query 'Einstein's biography ?', When EQ was considered in second document ranking, irrelevant relevant documents ' Einstein' was ranked first order.

5 Conclusion

Most IR systems do not consider lexical ambiguity of query terms, document terms and also index terms. And consider nouns and compound nouns kwyword by indexing word. Verbs and adjectives are useful indexing words. In this paper, we construct the Keyfact Network which is a kind of semantic network. Keyfact Network provides binary relationships such as BT, NT, RT, HP, UF, and FT. FT is a relationship noun-verb pair, noun-adjective pair, verb, adjective. At this time verb and adjective is restored to the original form. We resolved lexical ambiguity of query terms and improved the precision rate by considering EQ in the second document ranking.

In the future, we will expand the proposed Keyfact Network with a new ENCY, besides the encyclopedia have to experiment another test collection, and use the Keyfact Network in automatic indexing.

References

[1] Ellen M.Voorhees, "Query Expansion using Lexical-Semantic Relations", Proceedings of the Association for Computing Machinery-Special Interest Group on Information Retrieval, Dublin, pp.61-69, 1994.

[2] Robert Krovertz and W.Bruce Croft, "Lexical Ambiguity and Information Retrieval", Association for Computing Machinery Transaction on Information Systems, Vol. 10, No. 2, pp.115-141, 1992.

[3] Lesk M, "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone", Proceedings of Special Interest Group for Documentation, pp.24-26, 1986.

[4] Young S. Han, Young Kyoon Han, and Key-Sun Choi, "Lexical Concept Acquisition From Collocation Map", Proceedings of a Workshop Sponsored Group on the Lexicon of the Association for Computational Linguistics, Ohio, pp.22-31, 1993.

[5] Gerard Salton, *Automatic Text Processing*, Addison-Wesley Publishing Company, New York, pp.229-271, 1989.

[6] Roy Rada and Judith Barlow, "Document Ranking Using an Enriched Thesaurus", Journal of Documentation, Vol. 47, no. 3, pp.240-253, 1989.

[7] Yih-Chen Wang and James Vandendorpe, Martha Evens, "Relational Thesauri in Information Retrieval", Journal of the American Society for Information Science, Vol. 36, No. 1, pp.15-27, 1989.