A General Logical Approach to Inferential Information Retrieval

Jian-Yun Nie
Département d'Informatique et Recherche opérationnelle,
Université de Montréal
C.P. 6128, succursale Centre-ville
Montreal, Quebec
H3C 3J7 Canada
e-mail: nie@iro.umontreal.ca

Abstract

This paper describes a general approach to inferential information retrieval (IR). Different from the classical IR, in which the correspondence between a document and a query is estimated from a direct comparison of keywords included in them, an inferential approach also tries to infer indirect relevance relationship using a set of knowledge. A piece of knowledge is represented as an uncertain logical implication. It is converted from a thesaurus relation using relevance feedback. The experiments show that the approach is effective in practice.

1. Introduction

The purpose of Information Retrieval (IR) is to find relevant documents to a query. As documents are usually written in a natural language (we are concerned here with text retrieval), a document analysis is required in order to identify the contents of the documents, and to create an internal representation for them. When a query is written as a free sentence, the same process is also required for query analysis. Classical approaches to document analysis try to identify a set of important words from a document. The internal representation of the document is based on these words. In order to determine if a document is relevant to a query, the calculation of similarity usually depends on word sharing between the document and the query representations.

In other words, relevance estimation is based on a direct word matching between the document and the query.

It is known that a relevant document does not always contain the same words as the query. For example, a document about "unix" may be relevant to a query about "operating system". Yet the words "operating system" may be absent in that document. Therefore, the direct matching is unable to identify all the relevant documents. In order to broaden the coverage of a retrieval operation, an *inferential approach* is needed. If we call the result of the direct matching as the *direct relevance relationship*, then by an inferential approach, we mean an approach that is able to deduce an *indirect relevance relationship* between a document and a query such as in the case of "unix" vs. "operating system".

There have been a great number of studies trying to endow an IR system with the inferential capability. Different resources have been used, ranging from thesauri to statistical co-occurrences. However, in most cases inference is so diluted among many empirical measures that the inferential characteristic becomes unclear. In this paper, we intend to describe a logical approach for inferential IR. Logic is chosen as the pillar of the approach because inference is above all a logical operation. This will make the inferential characteristic very clear and general in the approach.

This paper considers inference as one of the basic operations in IR. In order to contrast it with classical approaches, we will first give a brief description of the latter. Some previous approaches close to inferential IR will be re-expressed from an inferential point of view. Then a general but simple logical framework will be presented. The underlying method derives new queries using a set of knowledge. This set of knowledge is adapted from a thesaurus to a particular application using relevance feedback. Finally some experimental results will be

described which show the effectiveness of the approach in practice.

2. Classical approaches to IR

2.1. Identification of descriptors and their measurement

As a document is usually written in a natural language, it cannot be compared directly with a query to estimate its relevance. An internal representation has to be created. This is a representation that can be directly manipulated by a computer. For example, one can represent a document as a Boolean expression of words, or as a vector of words. The internal representation is created by an *indexing* process.

The goal of indexing is twofold. 1) It identifies the most important concepts described in the document; and 2) it measures the importance of each concept in the document. The identification of *concepts* is difficult. Concepts are semantic entities. In order to identify them we have to have a conceptual or semantic model. Such a model does not exist in practice. Therefore, concepts are usually downgraded to *words*.

To identify the important words, various methods have been used in previous approaches. We will describe several of them in the following subsections.

Statistical method

In this method, only the frequency of word occurrences is considered. A word that appears very often in a document is considered as denoting an important concept in the document. It is thus considered as a representative of that concept. On the other hand, we also observe that many frequent words tend to be frequent in many documents. These words do not allow us to distinguish a document from the others. Therefore, they should not be assigned with a high value

of importance. This second aspect is also called *discriminative value* (Salton and McGill 1983). So the common practice is to combine the above two factors together in choosing and measuring the important words for a document. The *tf*idf* scheme is the most widely used weighting method (Salton and McGill 1983). This scheme determines the weight w_t of a word *t* according to its local frequency in the document (or *tf* - term frequency) as well as its distribution in the entire document collection (*idf* - inverted document frequency). One of the *tf*idf* formulas is as follows:

$$w_t = [\log(f(t, d)) + 1] * \log (N/n)$$

where $f(t, d)$ is the frequency of the term *t* in the document *d*, *N* is the total number of documents in the collection, and *n* is the number of documents including *t*. The part $[\log(f(t, d)) + 1]$ is derived from the term frequency $f(t, d)$, and $\log (N/n)$ is what we call *idf*.

Morphological analysis

It is observed that related words in European languages often only slightly change in form because of conjugation, agreements in number and tense, and so on. Such difference has little incidence in meaning. Therefore, a morphological analysis is often used in order to eliminate the difference in word form. The process is often called *word stemming* because it is often limited to eliminating the termination of words. For example, the word "information" may be transformed to "inform", so are the words "informative", "informed", and so on. Several stemming algorithms have been developed for this purpose. Two of them are the Porter's algorithm (Porter 1980) for English and an algorithm for French (Savoy 1994). The former eliminates terminations of words in several steps, using solely word morphology. The second method also uses a dictionary to determine the possible grammatical category of a word, and the possible stemming. Some other methods choose to transform words into a standard form (citation form). This latter approach has

been used in cross-language IR (Nie et al. 1999).

Syntactic analysis

As one may doubt, words are not the best descriptors of concept. Terms (especially compound terms) are often better descriptors. In an attempt to find more accurate concept descriptors, a number of approaches have tried to identify word couples, or noun phrases as document descriptors. In theory, we would think that such terms might provide more accurate representation of the document's contents. In practice, however, the obtained representation is merely better than single words. The problem is mainly due to the difficulty to identify the correct terms from a text. Usually, a set of syntactic patterns such as (NOUN NOUN) or (NOUN PREP NOUN) is used for the identification. Along with the correct terms, wrong groups of words (non-terms) are also identified. For example, from “data base system”, not only the correct term “data base” is identified, but also “data system” and “base system” because they also fit in the (NOUN NOUN) syntactic pattern. Despite the numerous studies devoted to this problem, there is no effective way to distinguish terms from non-terms from a syntactic point of view. As a consequence, the syntactic identification of compound terms is used in only a few IR systems, and its effectiveness is still to be proven (Fagin 1988).

There are still many other methods that have been developed in order to improve the identification of descriptors of concept. However, few of them have proven to be effective in practice.

Once a set of descriptors identified, the *tf***idf* weighting scheme or a variant of it is used for their measurement.

In the following discussions, we will use words or terms to refer to descriptors. By words, we mean single words used as descriptors. Terms may be single words or compounds (formed of

more than one word).

2.2. Internal representation and relevance estimation

Once a set of words (or terms) has been identified from a document (or a query) and their importance measured, an internal representation may be created for it. The two most used representation models are Boolean model and vector space model (Salton and McGill 1983).

Boolean model

In this model, a document is represented as a set of (possibly weighted) words. A query is usually a Boolean expression of words input directly by the user. Or it may be a Boolean expression converted from a user's query in free sentence. In this case, either AND or OR is used as the default operator to connect words from the sentence so as to create a Boolean query. For a document to be an answer to a Boolean query, an evaluation process is used to determine the relationship between them. One of the evaluation methods is as follows:

$R(d, t) = w_t$ (where w_t may be a weight obtained from the *tf*idf* weighting or a binary value);

$R(d, q_1 \wedge q_2) = \min(R(d, q_1), R(d, q_2))$;

$R(d, q_1 \vee q_2) = \max(R(d, q_1), R(d, q_2))$;

$R(d, \neg q_1) = 1 - R(d, q_1)$.

Another often-used method replaces the evaluations of conjunctive and disjunctive queries by the following:

$R(d, q_1 \wedge q_2) = R(d, q_1) * R(d, q_2)$;

$R(d, q_1 \vee q_2) = R(d, q_1) + R(d, q_2) - R(d, q_1) * R(d, q_2)$;

The documents that have the highest relevance degrees with the query are presented to the user as the retrieval result.

Vector space model

In vector space model, a document, as well as a query, is represented as a vector of weights. A vector space is determined by all the index words selected from the entire document collection. A value in a document (query) vector denotes the importance of the corresponding word in that document (query). In other words, given a vector space as follows:

Vector space: $< t_1, \quad t_2, \ ..., \quad t_n >$

A document and a query may be represented as the following vectors of weights:

$$d \rightarrow \quad <w_{d_1}, \quad w_{d_2}, ..., w_{d_n}>$$

$$q \rightarrow \quad <w_{q_1}, \quad w_{q_2}, ..., w_{q_n}>$$

where w_{d_i} and w_{q_i} are the weights of t_i in document d and query q. Query matching involves measuring the degree of similarity $sim(d, q)$ between the query vector q and each document vector d. The following calculation of similarity (Cosine formula) is among the ones often used in IR:

$$sim(d, q) = \frac{\Sigma_{i=1,n}(w_{d_i} * w_{q_i})}{[\Sigma_{i=1,n}(w_{d_i}{}^2) * \Sigma_{i=1,n}(w_{q_i}{}^2)]^{1/2}}$$

Again, the documents with the highest degrees of similarity are the answers to the query.

2.3. About the classical approaches

We can observe in all the classical approaches that a document should contain the same words (at least in part) as the query to be selected by the system. This is clear in the two models described above. In Boolean model a selected document should contain both conjuncts (or at least one of the elements in disjunction) in order to answer a query in conjunctive (or disjunctive) form. In vector space model, the more a document shares words with the query and in the same proportion of weights, the higher its similarity is to the query.

We see here a gap between the goal of IR and its realization methods: The goal of IR is to retrieve documents of certain meaning; whereas it is implemented as that of retrieving documents containing the same words. Although it is not formulated as such, the following assumption is implicitly made in the classical models: there is a strict correspondence between words and meanings. This assumption is clearly incorrect: a meaning may be expressed by different words, and a word may express different meanings in different contexts. This wrong assumption is exactly the limitation of the classical approaches to IR. How to overcome this limitation is the objective of many current research projects.

This problem concerns two main aspects: 1) multiple meaning of a word; 2) multiple expression of a meaning. The first problem is related to word disambiguation. A word's meaning has to be determined in dependence of its context of utilization. This problem has been dealt with in linguistics as well as in IR (Voorhees 1993, Voorhees 1994). Our work is not directly related to this aspect. Rather, we will be concerned with the second aspect in the inferential approach: Given a query expressed by some words, we want to determine if some other words may also express the same meaning. That is, we intend to infer new words from those given by

the user, and to form a new query from them. For example, from a query on "operating system", we would infer the word "unix" and create the new query "operating system $\vee$ unix". In such a way, the documents retrieved with the new query have an indirect relevance relationship with the user's query (in contrast with the direct relevance relationship in the classical methods). In some way, the central problem is to make use of the relationships between words in order to determine which words are related. This is not a trivial task because of polysemy. A word may be related to another word in one context but not in another. If we add all the related words into the query regardless to the application area, a lot of noise (irrelevant documents) will be retrieved. So, it is important to first select the relationships between words that are appropriate to the application domain and to the user's information need, before they are applied.

In the following sections, we will describe several existing approaches related to inferential IR. We will see that the idea of inferential IR has existed for a long time. In our description of these approaches, we will re-express them from an inferential point of view. This will allow us to highlight the common characteristics of inferential approaches.

3. Previous approaches to inferential IR

3.1. Query expansion - a general form of inferential approach

Although in its initial form, query expansion is not described as an inferential approach, it does possess the main characteristic of an inferential approach, that of inferring indirect relevance relationship.

Query expansion works as follows: Given an initial query of the user, some new related words are added and this forms a new query. The addition of the new words extends the original query so that it has a wider coverage than the original query. Therefore, even if a document does

not use the same words as the original query, it may still be judged to be relevant if it contains the words that are added through query expansion. As a consequence, more relevant documents may be retrieved, and the recall ratio be increased. The key problem is to add the appropriate words. Otherwise, the new query will depart from the original query (in meaning). So an important question is what words should be added. Another important question is how they should be integrated into the new query.

How are new words integrated into the query?

Let us first examine this problem with respect to the two models described earlier. In Boolean model, the added words are put into disjunction with the original query words. For example, if t is a word in the original Boolean query and t_1 is a related term to it, then t_1 is put into disjunction with t in the new query. In some cases, the added term is assigned an equal important to the original term t. Thus, t is replaced by $(t \vee t_1)$. In other cases, the added term is assigned a lesser importance. So t is replaced by $(t \vee t_1^{\alpha})$ where $\alpha \leq 1$. The factor α has been assigned different functions. In some systems, it plays the role of a multiplication factor of the similarity obtained with t_1. That is, if a document's similarity to t_1 is v, then its similarity to t_1^{α} is $(\alpha * v)$. In some other systems, it is an upper bound of the similarity, i.e. the similarity to t_1^{α} is $min(\alpha, v)$.

The effect of query expansion is to enable the system to retrieve, besides the documents containing the word t, also the documents containing the word t_1. If the added words are truly related to the original query, the recall ratio may be increased. On the other hand, if the new words added are not truly related to the query, the precision ratio will decrease.

In vector space model a related word is added into the corresponding vector dimensions of the query vector if they do not exist in the original vector. If it exists, its weight is increased by a certain factor. The effect of query expansion in vector space model is similar to that in Boolean model. However, the new word is not considered as an alternative of the original word, but as a supplement to it.

Which words are added? - The use of thesauri

Intuitively, the added words have to have a strong semantic relationship with the existing words in the original query. That is, a new word should describe a concept which is strongly related to the concepts described in the original query. There are two ways to determine these words: the system may select the words interactively with the user; or it may do it automatically. Systems working in the first manner usually possess a thesaurus that stores a set of possible relationships between terms/words. For example, for each word/term, a thesaurus may store a set of synonyms, more specific and more general words/terms. If a term/word is included in a user's query, the system will suggest the related terms/words to the user. The user selects those that are related to his/her information need to be added into the query. This expansion method relies on intensive interactions with the user. In practice, this is often a heavy burden to users. It is only used in domains where specialized thesauri are available, and the users are ready to make efforts to cooperate with the system to find relevant documents.

The second method - the automatic query expansion - has been studied extensively in the last two decades. This approach also relies on a thesaurus (or pseudo-thesaurus). An automatic process first tries to identify the most closely related words from the thesaurus and then adds them to the query. Among the thesauri used in such an approach, there are classical thesauri that

are established manually, or pseudo-thesauri that are established automatically. A manual thesaurus usually contains a set of semantic relationships between words or terms in a specialized domain, or in general domains. Usually automatic query expansion consists of selecting the related words/terms that are linked with the original query words through some types of relationship that are judged to be "strong" relationships. Typically, the *is_a* relationship is one of them. It is also observed that the so-found new terms may have very different meanings than the original terms. Therefore, it is a common practice that the related terms are weighted less than the original terms. In addition, the longer the relationship path one has to traverse to link the two terms, the lower the new term is weighted in the new query. This approach is used in (Rada et al. 1991; Salton and Buckley 1988).

On the other hand, automatically constructed thesauri are usually based on statistics on word co-occurrences (Rijsbergen 1977): the more two terms co-occur in the same context, the stronger they are considered to be related. Context may vary from document, paragraph to sentence. While this kind of thesaurus may help users to some extent, their utilization in IR shows that their impact on the global effectiveness is limited (Sparck-Jones 1991). In (Grefenstette 1992), it is also shown that when queries are expanded using term co-occurrence information, a worse system effectiveness is obtained.

The limited effect of statistical thesauri on IR is due to several reasons. First, real semantic relations, in particular, synonymy, can be hardly identified statistically. In fact, words very similar in meaning tend to repulse from each other in continuous portions of text (Sinclair 1991). As an example, one may think about "tumor" and "tumour". They are rarely used together. It is also a general advice in scientific writing to avoid as much as possible using different words for the same meaning. Therefore, statistical techniques can only identify related

words that may appear in the same context, but not the strong relationship of synonymy.

Second, statistical relationships usually link terms of similar frequency of occurrences in document collection. They have similar degree of generality or specificity to the application area. Adding such a related word into a query does not make the original query more specific or more general. Therefore, it does not bring much new information to the query. In addition, as users tend to use very frequent words in their queries, the added words also are frequent words. They have very poor discriminative value to distinguish a document from the others. Therefore, the effect of such a query expansion is limited (Peat and Willett 1991).

Third, statistical relationships may often relate two independent words together. Expanding a query by such a relationship may only increase noise in the answer. To cope with this problem, it has been suggested that the co-occurrence context should be confined (Grefenstette 1992, Hearst 1992, Hindle 1989). Instead of co-occurrence within documents or paragraphs, more specific syntactic contexts have been used, for example, subject-verb or adjective-noun contexts. However, the syntactic restriction may not bring much improvement. Grefenstette (1992) reports that, although improvement has been observed, it is marginal.

The use of statistical relationships in the previous tests was often due to the lack of suitable manual thesauri. Statistical thesauri were thought of to be a reasonable replacement of manual thesauri. However, despite the numerous studies devoted to it, there is no effective way to find good statistical relationships that may significantly impact IR systems.

Either in using a manual thesaurus or a statistical thesaurus, Qiu and Frei (1993) observed that one usually adds words that are related to individual words of the original query, instead of to the entire query. In fact, the relationships between words have been usually set up in a general context. They may not be suitable for a query that concerns a particular area.

Therefore, they suggest that query expansion should add words that are strongly related to the entire query. In such a way, the added words are more specialized to the particular area of the query, and less noise will be introduced. Qiu and Frei showed that the approach brings a significant increase in performance. In a general way, query expansion is faced with the problem of selecting appropriate new words. A general thesaurus can only provide related words/terms in general area. It is inappropriate to use it directly to the particular area of the query.

3.2. Relevance feedback and pseudo-relevance feedback

Another problem in IR is that the user's query is often a bad description of the information need: important words are missing, ambiguous words are used, and so on. In this case, even if a thesaurus is used, it is unlikely that the missing important words may be added.

An alternative to query expansion based on a thesaurus is to use the user's relevance feedback. After the system has retrieved a set of documents, the user is asked to judge some of them, indicating whether they are relevant. Based on these indications, the system can have a more precise idea on what the user's intention is. It is assumed here that the user is interested by the documents similar to the ones that are judged relevant, and is not interested to those judged irrelevant. By incorporating the words found in the relevant documents (or increasing their weights), and eliminating those in the irrelevant documents (or decreasing their weights), it is expected that the new query is closer to the user's intention. The typical query reformulation using relevance feedback is that of Rocchio (Salton and McGill 1983):

$$New\_Query = \alpha * Old\_Query + \beta * R - \gamma * NR$$

where R and NR are the centroids of the set of relevant and irrelevant documents judged by the user; α, β and γ are factors that determine the importance of the old query, the relevant and irrelevant documents to the new query. As we can see in the formula, the new query becomes closer to the relevant documents and more distant from the irrelevant documents.

An improved process is called *incremental relevance feedback*. The difference with the above process is that the system does not wait to have all the relevance feedback information from the user to re-formulate the new query. Instead, as soon as a single judgment is made, a new query is created and evaluated. It has been shown that this modified process allows the user to obtain the same number of relevant documents more quickly. However, much more re-evaluations are required.

Although relevance feedback has proven to be an effective way to improve IR performance, it is rarely used in practice. This mechanism has been incorporated into several online search engines, but few users actually use it. We think that an important reason is the short-term effect of a relevance feedback. A user has to make great efforts in judging documents, but the judgements only have an effort on a single query. Once a new query is input, new judgements have to be made. From the user's point of view, it simply does not worth the efforts. However, we think users are ready to make the efforts if the effect is permanent. Therefore, we will suggest a way to adjust the system's knowledge according to relevance feedback.

3.3. Using probabilistic inference in IR

IR has a long history of using probability theory (Bookstein 1983, Maron and Kuhns 1960, Rijsbergen 1979, Robertson et al. 1982). The earliest model is called independent probabilistic model because it is assumed that terms are stochastically independent, i.e. the probability of observing a term is the same with or without the presence or absence of other

terms. Under this assumption, the probability $P(R_q|d)$ - the probability of finding relevant documents for the query q given the document d - may be estimated as follows.

First, the Bayes theorem is used:

$$P(R_q|d) = \frac{P(d|R_q) * P(R_q)}{P(d)}$$

In this formula, $P(d)$ is assumed to be a constant that is dependent on the document collection. $P(R_q)$ is another constant that is dependent on the query. The key problem is to estimate $P(d|R_q)$ - the probability of having the document d given all the relevant documents for the query. It is in the estimation of this probability that the independent assumption is used. A document d is represented by a set of events, which are the presence or absence of each term. Using x_i to denote the value (presence or absence) of the term t_i in the document, and x_i=1 and x_i=0 to represent the events of presence and absence of the term, $P(d|R_q)$ becomes the following:

$$P(d|R_q) = \prod_i P(x_i=1|\ R_q)^{x_i}\ P(x_i=0|\ R_q)^{(1-x_i)}$$

In order to estimate $P(x_i=1|\ R_q)$ and $P(x_i=0|\ R_q)$ - the presence or absence of a term among the relevant documents, one can use a set of samples of document whose relevance is judged (Rijsbergen 1979).

In IR, one does not have to obtain a precise value for $P(R_q|d)$. Instead, we are interested in ordering documents according to their relevance estimation. Therefore, one often uses the comparison of $P(d|R_q)$ and $P(d|NR_q)$ (where NR_q represents the set of no-relevant documents) to estimate if a document should be retrieved, and how it should be classified in the result

It is clear that the assumption of independence is incorrect. For example, when we

observe the word “algorithm”, we have a much higher probability to observe the word “computer” than observing it in general. One of the earliest attempts to consider dependency between terms tries to extend the independent model by incorporating a term dependence tree (Rijsbergen 1979) as follows:

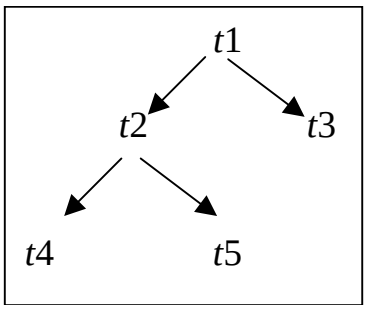

Figure 1. Binary dependence tree.

In this model, the dependency of a term to another term, e.g. $P(t_2|t_1)$, is considered, but not the dependency of several other terms such as $P(t_2|t_1, t_3)$. This dependence endows the model with a certain, however limited, inferential power. As in reality the term dependency does not form a tree, it is necessary to make a selection among all the dependencies to form a tree. The method suggested in (Rijsbergen 1979) is that the tree should represent the strongest dependencies. However, this is difficult to determine, and the effect of incorporating such a tree is very limited. In fact, when we choose to use only a part of the dependencies, we create a very partial view of the real dependencies among terms. This view may be strongly distorted. Therefore, its effect is not always positive.

More recently, Bayesian networks (Pearl 1988) have been used in IR with great success (Turtle and Croft 1990). This method is based on a pre-established inferential structure, divided into several layers (document, text, term, concept, query and information need) as follows:

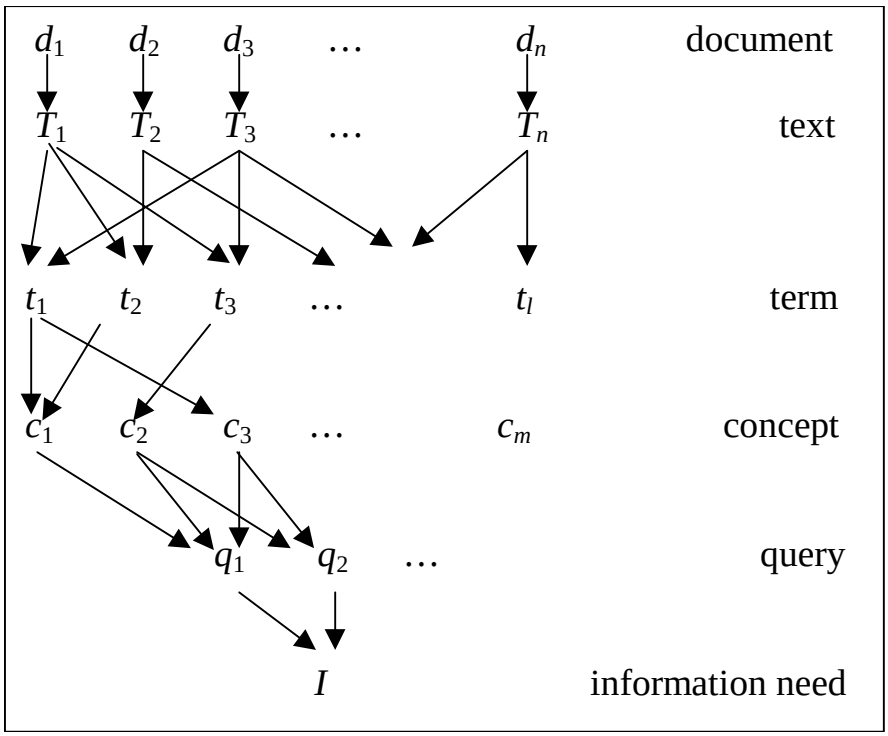

Figure 2. Bayesian network

The distinction between document and text is due to the fact that a modern document may be represented by different texts. In this first model, however, it is assumed that each document corresponds to only one text. At the bottom, queries are different from the information need. An information need may be expressed by various queries. In the middle, a text is represented by a set of terms. Each term may correspond to one or several concepts. A query includes a set of concepts.

As a constraint, elements in one layer may be connected to the elements of directly

adjacent layer, but no connection is allowed among elements of the same layer.

The connections in the network are associated with probabilities. For example, in the figure above, c_3 only depends on t_1, so the connection between them is associated with the probabilities $P(c_3|t_1)$, $P(\neg c_3|t_1)$, $P(c_3|\neg t_1)$ and $P(\neg c_3|\neg t_1)$. As c_1 depends on both t_1 and t_2, the probability of c_1 and $\neg c_1$ should be conditioned by the combinations of t_1 and t_2 (i.e. (t_1, t_2), $(t_1, \neg t_2)$, $(\neg t_1, t_2)$ and $(\neg t_1, \neg t_2)$). In general Bayesian networks (Pearl 1988), the key problem is the setting of dependency between elements, and the estimation of their probabilities. In the above model used in IR, the connections have been much simplified. The combinations of elements on the right side of the conditional probabilities have been limited to Boolean combinations.

In Bayesian networks, there are two basic operations: forward probability spreading from the top to the bottom, and the backward probability revision. Applied to IR, the first operation allows us to estimate the probability of satisfying an information need by each document. Then the documents with the highest probability are selected. The second operation allows us to re-adjust the probabilities associated to the connections. This second operation is important because the first setting of the connections may not be always appropriate. In theory this operation may well be realized with the help of relevance feedback. However, the model proposed in (Turtle and Croft 1990) does not include this operation.

Although the Bayesian networks are able to incorporate quite complex relations, the banning of dependence among elements of the same layer is still too strong. For instance, for the term and concept layers, the independence assumption implies that one has to determine a set of elementary terms and concepts that are independent from each other. This is an assumption too strong in practice.

3.4. Logical IR

Inference is above all a logical operation. Inferential approaches to IR have also been formulated in logic terms. The classical Boolean model is a particular case of a more general Boolean logic framework. This framework may be described as follows:

A document *d* is represented as a set of terms, or equivalently as a conjunction of terms. Each term corresponds to an atom. A query is a Boolean expression of terms. The relevance of a document represented by *d* to a query represented by *q* is determined by the logical implication $d \rightarrow q$. A logical system is characterized by a set of logical sentences (or its closure). If we represent it by *K*, then the relevance of *d* to *q* with respect to this system is expressed as $K \vdash d \rightarrow q$. If we have $K \vdash d \rightarrow q$, the document is said to be relevant. If we cannot prove $K \vdash d \rightarrow q$, it is irrelevant.

What is interesting to observe in this model, in comparison with the Boolean model described earlier, is the appearance of *K*. That is, the system's evaluation is characterized by a set of logical sentences. We call this set of sentences the *system knowledge*. It is this set of knowledge that is the basis of inference in relevance estimation. For example, if *K* contains the sentence $(a \rightarrow b)$, then given a document talking about *a*, the system is able to judge that it is also relevant to a query on *b* using a simple inference. One can imagine that more complex knowledge may be incorporated in such a system.

Although the above description seems to be very natural, it is surprising to see that the model is not widely used entirely in IR. More often, one is limited to the classical Boolean model, which is equivalent to the above logical model with an empty *K*. In this case, a selected document should contain the same terms as the query. A possible reason to this limitation is that the importance of knowledge in IR has not been fully recognized until mid-1980s (Croft 1987).

Another problem is related to the difficulty to set up knowledge in such a system. The model itself is also restricted by a strict binary evaluation: It can only tell if a document is relevant or not, but is unable to associate a degree of relevance.

The strict binary evaluation may be smoothed by different means. There have been several works using fuzzy set theory (Buell 1982; Radecki 1979; Waller and Kraft 1979) to replace the strict evaluation in classical logic. Unfortunately, inference has not been a concern in these attempts.

There are also attempts in developing a suitable logic for information retrieval to cope with inference. The idea proposed by van Rijsbergen (Rijsbergen 1986, Rijsbergen 1989) attracted much attention. It is suggested that relevance be expressed as a non-classical implication $d \rightarrow q$. Its degree of certainty is determined by a function $P(d \rightarrow q)$. To evaluate $P(d \rightarrow q)$, van Rijsbergen proposed the following *uncertainty principle*:

> Given any two sentences x and y; a measure of the uncertainty of $y \rightarrow x$ relative to a given data set, is determined by the minimal extent to which we have to add information to the data set, to establish the truth of $y \rightarrow x$.

This proposal has triggered a series of investigations on logical IR (Bruza and Huibers 1994; Crestani and Rijsbergen 1995; Lalmas 1996; Nie, Brisebois and Lepage 1996). These studies investigated the problem of relevance in IR from different logical points of view. Although the proposed models stand from a theoretical point of view, they are often difficult to implement. In this paper, we use a simple logical framework. The resulting IR model is much easier to implement.

4. A general logical approach to inferential IR

4.1. The model

As in the logical model described in Section 3.4, the relevance of a document d to a query q with respect to a system K is expressed as follows:

$$K \mid\!\!- d \rightarrow q.$$

We will denote the degree of certainty of this formula by the function $P_K(d \rightarrow q)$ (or simply $P(d \rightarrow q)$ because P is defined for a single system). Inspired from van Rijsbergen's uncertainty principle, we can formulate the following two approaches:

Approach 1. In order to estimate the degree of certainty of $K \mid\!\!- d \rightarrow q$, we must identify all the document descriptions d' related to d. The degree of certainty of $K \mid\!\!- d \rightarrow q$ is determined by both the degree of relatedness of d' to d and the degree of certainty of $K \mid\!\!- d' \rightarrow q$.

Approach 2. In order to estimate the degree of certainty of $K \mid\!\!- d \rightarrow q$, we must identify all the query expressions q' related to q. The degree of certainty of $K \mid\!\!- d \rightarrow q$ is determined by both the degree of relatedness of q' to q and the degree of certainty of $K \mid\!\!- d \rightarrow q'$.

As these approaches are based on modifications of the document description and the query description respectively, we call them *document-driven* and *query-driven* approaches. In the following discussions we will focus on the query-driven approach because this approach is easier to implement in IR and is closer to the traditional query expansion.

In fact, these approaches are based on exactly the same principle as the transitivity of classical logic implication:

$$A \rightarrow B \wedge B \rightarrow C \ |\!-\ A \rightarrow C$$

The query-driven approach can be rewritten as follows to have exactly the same form:

$$d \rightarrow q' \wedge q' \rightarrow q \ |\!-\ d \rightarrow q$$

It means: if there is a new query *q'* such that the new query implies the original query, and that the new query is satisfied (implied) by a document, then we can say that the original query is also satisfied by the document.

As *q'* may be any query expression, we can re-write the above deduction as follows:

$$\vee_{q'} (d \rightarrow q' \wedge q' \rightarrow q) \ |\!-\ d \rightarrow q$$

Interpreting this formula in a context which involves uncertainty, we can obtain the following evaluation:

$$P(d \rightarrow q) = P\big(\vee_{q'} (d \rightarrow q' \wedge q' \rightarrow q)\big) \qquad (1)$$

We notice that the right side of the equation (1) includes two factors:

- $P(d \rightarrow q')$ measures the *degree of (direct) satisfaction* of a query *q'* to the document *d*.
- $P(q' \rightarrow q)$ measures the *degree of relatedness* of the query *q'* to the original query *q*.

Therefore, the evaluation of $P(d \rightarrow q)$ is determined by both these factors, and for every possible *q'*.

In order to evaluate the right side of the formula, we have to define evaluation methods for logical conjunction and disjunction. In fuzzy set theory, many evaluation methods have been proposed for them. A general form - known as *triangular norm* Δ was suggested by (Dubois and Prade 1984) for the evaluation of conjunction. A triangular norm Δ: [0,1] x [0,1] $\rightarrow$ [0,1] is a

function that verifies the following conditions (where $x, x', y, y', z \in [0,1]$):

1. $\Delta(x, y) = \Delta(y, x)$;

2. $\Delta(x, \Delta(y, z)) = \Delta(\Delta(x, y), z)$

3. If $x \geq x'$, and $y \geq y'$, then $\Delta(x, y) \geq \Delta(x', y')$.

Correspondingly, a disjunction is evaluated by a triangular co-norm ∇ which is defined as follows:

$$\nabla(x, y) = 1 - \Delta(1-x, 1-y).$$

The function *min* is a triangular norm. Its co-norm is *max*. Multiplication of real numbers is another triangular norm. Its co-norm is $(x + y - x*y)$. These two sets of functions are among the most used functions for logical operators in fuzzy set theory.

Using a triangular norm Δ and its co-norm ∇, we can further develop equation (1) as follows:

$$P(d \rightarrow q) = \nabla_{q'} [\Delta(P(d \rightarrow q'), P(q' \rightarrow q))]$$

As we stated earlier, we can define the evaluation of a query of the form A^{β} with respect to a document d as $P(d \rightarrow A^{\beta}) = \Delta(P(d \rightarrow A), \beta)$. Therefore, the above equation may be rewritten as follows:

$$P(d \rightarrow q) = \nabla_{q'} [P(d \rightarrow q'^{P(q' \rightarrow q)}]$$

$$= P(d \rightarrow \vee_{q'} q'^{P(q' \rightarrow q)}) \qquad (2)$$

What is interesting in the above equation is that the whole inferential approach becomes a derivation of a new query $\vee_{q'} q'^{P(q' \rightarrow q)}$. This new query is evaluated directly with respect to a document. So the entire inferential power of the approach is embedded in the derivation of such

a new query. This make the approach very similar to the usual query expansion approach.

Notice also that it is not necessary that all the q' included in the new query should be directly derivable from the original query q. A q'' may be derived from q via some other q' such as shown in the following figure:

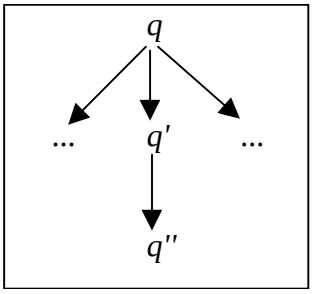

Figure 3. Indirect derivation of a query

In this case, the uncertainty involved in both derivation steps (i.e. from q to q' and from q' to q'') has to be taken into account for q''.

The derivation of a new query form is, however, not arbitrary. A query q' is to be considered only if it has some relationship with the original query, i.e. $P(q' \rightarrow q) > 0$. The finding of such q' should be guided by a set of knowledge. For example, if we consider that a document about *processor* is also relevant to a query on *computer*, then from a query on *computer*, we can derive a new query about *processor*. This means that if a document matches the derived query (*processor*), the document is also relevant. However, we cannot always expect that the document matching *processor* is fully relevant to a query on *computer*. In general, a document matching an alternative query is only partly relevant to the original query. So, we can associate a value to each relationship between terms to denote their strength of relatedness. This strength is, in fact, the simplest form of degree of relevance: if the strength is high, then the document matching the derived query has also a high chance to be relevant to the original query. Without loss of generality, we assume that the set of knowledge of an IR system is expressed as a set of

uncertain logical implications between terms such as $a \rightarrow_{\beta} b$, where $\beta \in [0,1]$ denotes the strength. In the next section we will discuss how such a set of knowledge may be obtained.

The query derivation is performed as follows: For an original query which involves a term a: $q = (\ldots\ a\ \ldots)$, if the set of knowledge contains $a \rightarrow_{\beta} b$, then the original query can be derived to $q' = (\ldots\ (a \vee b^{\beta})\ \ldots)$. That is, each appearance of a is replaced by $(a \vee b^{\beta})$. This process is repeated for every term of the query, and recursively. If a new term b is to be expanded through the relation $b \rightarrow_{\beta'} c$, the above q' becomes:

$$q'' = (\ldots\ (a \vee (b \vee c^{\beta'})^{\beta})\ \ldots)$$

$$= (\ldots\ (a \vee b^{\beta} \vee c^{\Delta(\beta, \beta')} \ldots)$$

It is easy to check that the obtained query expression is equivalent to $\bigvee_{q'} q'^{\,P(q' \rightarrow q)}$ in equation (2).

The problem remained is how the system's knowledge may be obtained and expressed as uncertain logical implications. We will examine this problem in the following section.

5. Adaptation of a thesaurus to a knowledge base - A case study

It is not easy to create the system's knowledge manually. There is currently no knowledge base of the required form that is ready to be used in IR. However, there are other kinds of knowledge bases. Typically, one uses a thesaurus as a knowledge base in IR. Such a thesaurus is different from our knowledge base with respect to the following two aspects:

1) A thesaurus is usually a general knowledge base which contains relationships between terms for various application areas. What we need is a knowledge base that is tailored to a particular application area (e.g. computer science). Therefore, a selection or further

adaptation should be made before it is used in a particular application area.

2) The required knowledge base only contains one type of relationship (logical implication) and that each relationship is quantitatively measured. However, a thesaurus usually classifies relationships according to their semantic nature (e.g. synonymy, hypernymy), and no quantitative measure exists.

In order to take advantage of existing thesauri, we propose to adapt gradually a thesaurus to an application area through relevance feedback. The principle is as follows: If using a relationship to expand an original query leads to a higher performance, then the strength associated to that relationship may be increased. If on the contrary, the performance decreases, the strength of that relationship should also be decreased.

The assumption behind this approach is that an application area (or a set of users with similar background) has a set of typical knowledge. The strength associated to each relationship may be appropriately set and adjusted using a set of typical queries and relevance feedback for them.

This assumption is reasonable, in particular, when the IR system is used in a specialized area. When IR is used in such an area, a set of common knowledge is generally used implicitly. The goal of our adjustment process is to make this knowledge explicit.

More precisely, for a given thesaurus relation between a and b, the strength β of the corresponding relevance relation $a \rightarrow_{\beta} b$ is determined as follows:

1) If a user's query contains a, the system carries out a tentative evaluation with an initial value β for the thesaurus relation, i.e. the query is expanded with b^{β}.

2) The user examines (possibly part of) the documents in the response and indicates if they

are relevant or not.

3) Then two alternative values to β are calculated:

$$\beta' = min[1, \beta*(1+\varepsilon)]$$

and $\beta'' = \beta*(1-\varepsilon)$.

where $\varepsilon \in]0,1[$ is the change scale. The query is evaluated again with β' and β''.

4) The value β for this relation is adjusted to the value which leads to the best answer, i.e. either it remains β or it changes to β' or β''.

In the third step, we use the *average precision* (Salton and McGill 1983) to measure the quality of a retrieval result.

Although we use relevance feedback as in traditional approaches to query re-formulation, the object revised is different: in our revision, we revise the system's knowledge. This revision has a long-term effect on query evaluation. We think that users are more interested in collaborating in such a revision process that in the classical relevance feedback process for each individual query.

Learning for a group of thesaurus relations vs. for individual relations

There are usually a great number of semantic relations in a thesaurus. If we try to revise each individual relation at the beginning, a great number of queries are required. In order to accelerate the process, we suggest to make a two-step revision: First, all the relations of the same type are assumed to have the same strength, and they are revised together. Second, individual relations are revised individually.

The purpose of the first step is to have a quick idea on each type of relation of the entire thesaurus. In a thesaurus, the criteria used to establish each type of relation may vary. For

example, in a single thesaurus, one may apply a strict criterion on synonymy, but a loose criterion on the "see-also" relation. In addition, as a thesaurus is built for purposes other than IR, some types of relation are useful to IR, whereas some others are not. This first step of revision also allows us to estimate the appropriateness of each type of relation to IR.

However, the relations of the same type do not necessarily have the same value for IR. Therefore, a second step of revision is required to fine-tune their strength.

6. Experiments

The approach has been tested on the CACM corpus which comprises 3204 documents published in the *Journal of the ACM*. A set of 50 queries have been manually evaluated (i.e. we know which documents are relevant to them). Document descriptions are obtained using an automatic indexing process based on document's title and abstract and using *tf***idf* weighting method.

Thesaurus and its utilization

The thesaurus Wordnet (Miller 1990), version 1.5, is adapted into a system knowledge base through relevance feedback. Wordnet contains a large set of human-defined relationships among English words and terms. The following table shows the types of relation it contains:

relationship	example
synonymy	computer ⁄ data processor
antonymy	big ⁄ small
hyponymy (A-KIND-OF)	tree $\Rightarrow_{\text{hyponymy}}$ maple
hypernymy (IS-A)	maple $\Rightarrow_{\text{hypernymy}}$ tree
meronymy (HAS)	computer $\Rightarrow_{\text{meronymy}}$ processor
holonymy (IS-PART-OF)	processor $\Rightarrow_{\text{holonymy}}$ computer

Table 1. Some examples of relations defined in Wordnet

A word (or term) sense is identified by a group of terms that are synonyms under this sense. Such a group is called a *synset*. A given word (or term) may be contained in several synsets, each corresponding to a sense. The synonymy relation is implicitly defined among all the terms in the same synset. The other relations are established between synsets. Here we give an example to illustrate the organization of Wordnet. We denote a synset by {...}, and a given type of relation by $\Rightarrow_{type}$.

The word "computer" is included in the following two synsets:

Sense 1:

{computer, data processor, electronic computer,
information processing system}
$\Rightarrow_{hypernymy}$ {machine}

Sense 2:

{calculator, reckoner, figurer, estimator, computer}
$\Rightarrow_{hypernymy}$ {expert}

That is, "computer" has two different meanings: one for a machine (sense 1) and another for an expert (sense 2). The synset after '$\Rightarrow_{hypernymy}$' is the hypernym synset of the given sense. $a \Rightarrow_{hypernymy} b$ means "*a* IS_A *b*".

The hyponymy relation (the reverse of hypernymy - A_KIND_OF) related to "computer" is as follows:

For sense 1:

computer $\Rightarrow_{\text{hyponymy}}$
{analog computer, analogue computer}
{number cruncher, number-cruncher}
{digital computer}
{pari-mutuel machine, totalizer, totaliser, totalizator, totalisator}
{tactical computer}

For sense 2:

computer $\Rightarrow_{\text{hyponymy}}$
{number cruncher, number-cruncher}
{statistician, actuary}

The meronymy (HAS) relation only exists for sense 1:

computer $\Rightarrow_{\text{meronymy}}$
{cathode-ray tube, CRT}
{chip, microchip, micro chip, silicon chip}
{computer accessory}
{computer circuit}
{busbar, bus-bar, bus}
{analog-digital converter}
{disk cache}
{diskette, floppy, floppy disk}
{hardware, computer hardware}
{central processing unit, CPU, C.P.U., central processor, processor, mainframe}
{keyboard}
{monitor}

Each relation between two terms is considered as an uncertain logical implication. For example, "computer $\Rightarrow_{\text{hypernymy}}$ data processor" is transformed to "computer $\rightarrow_{\beta 1}$ data processor" where β_1 is the degree of certainty of the implication. The higher β_1 is, the stronger the sense of "data processor" is related to that of "computer". Of course, β_1 should be set at an appropriate value according to the application domain. We will deal with the setting of this value in the next subsection. For the moment, we assume that each relation between terms is transformed to an uncertain logical implication. We will see how the relations will be used in

query expansion.

Suppose the initial query is q = 'computer', and relevance strength of all the terms in the first synonym synset of "computer" is β_1 and that of the terms in the second synset is β_2. Then the expanded query q' with synonymy relation alone is as follows:

$$q' = \text{'computer'}$$
$$\vee \text{'data processor'}^{\beta_1} \vee \text{'electronic computer'}^{\beta_1} \vee$$
$$\text{'information processing system'}^{\beta_1}$$
$$\vee \text{'calculator'}^{\beta_2} \vee \text{'reckoner'}^{\beta_2} \vee \text{'figurer'}^{\beta_2} \vee$$
$$\text{'estimator'}^{\beta_2}$$

This query may be further expanded with other types of relations from the term "computer".

We see that each term is added into the query as an alternative to the original term (i.e. connected with logical-or $\vee$). In addition, the new terms are associated with their degree of relatedness to the original term, reflecting how well a document satisfying a new term may also be good for the original term.

The query may be expanded with more related terms by applying the above expansion process several times. That is, a newly added term may also be expanded by other related terms. For example, for "calculator" we have the following relation:

$$\text{calculator} \rightarrow_{\beta 3} \text{calculating machine}.$$

Therefore, the term "calculator" in the above expanded query may be expanded to:

$$\text{'calculator'} \vee \text{'calculating machine'}^{\beta_3}$$

Thus $\text{'calculator'}^{\beta_2}$ in the expression becomes

$$(\texttt{'calculator'} \vee \texttt{'calculating machine'}^{\beta_3})^{\beta_2}$$

$$= \texttt{'calculator'}^{\beta_2} \vee \texttt{'calculating machine'}^{\Delta(\beta_2, \beta_3)}$$

In order to avoid expanding queries by too many remotely related terms, we can limit the expansion to a certain length: The application of all types of relation on the original query terms corresponds to the expansion of length 1. The application of the relations to the expanded terms leads to an expansion of length 2, ...

Compound terms are then represented as a conjunction of simple terms after expansion (this is because documents have been represented by single words through the indexing process). For example, 'data processor' will be replaced by (data $\wedge$ processor).

The expanded query may be very long, especially for a long inference. However, many new terms are added with very low weight, especially those added after several steps of expansion. Thus, by setting a threshold, the expanded query can be easily reduced to an acceptable length. In addition, many added terms do not correspond to any document (for example reckoner in the CACM collection). They will be removed.

Learning for different types of relations

We first used the assumption that the 50 queries are evaluated by experts having the same background (computer science and IR) and judgment criteria. The 50 evaluated queries are randomly separated into 5 groups of 10. Each group is used in turn as the set of test queries while the others are used as training queries for term relevance strength.

We first tested the learning for types of relations. At the beginning of each training process, the relevance strength 1 is attributed to each type of relation. Different values for the change scale ε have been tested. It is observed that with a too low value of ε, relevance strengths

change too slowly while with a too high value of ε, relevance strengths become unstable. The value 0.15 seems to give a good compromise between learning speed and stabilization for our case. The following figure shows the average evolution (the average of the 5 training processes) of relevance strength for each type of relation as the learning proceeds.

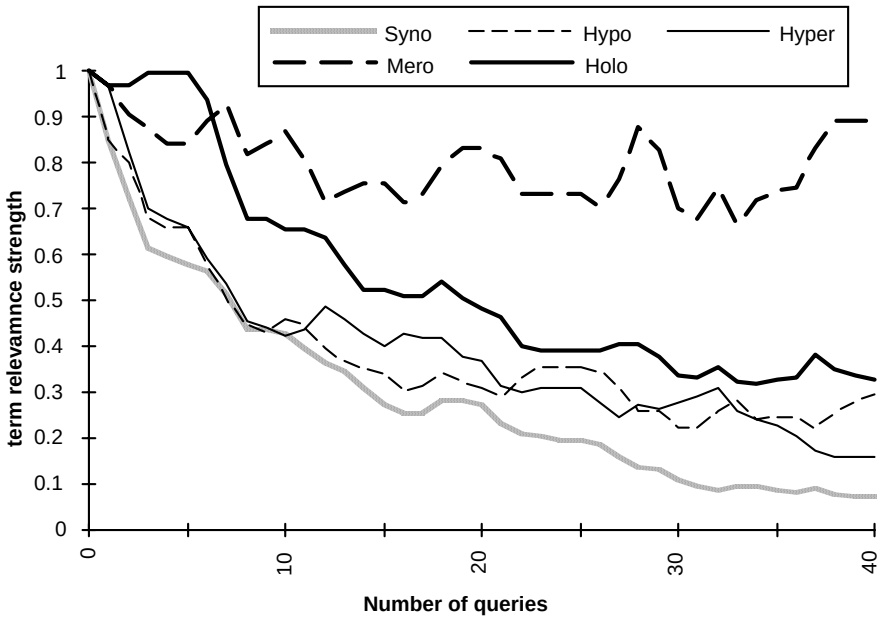

Figure 3. Evolution of relevance strength along with the leaning process (from (Nie and Brisebois 1996)).

It is surprising to observe that synonymy relations have the lowest strength. Intuitively, we would rather attribute the strongest strength to them. The reason for this is that Wordnet is thesaurus that covers many domains. People who built Wordnet are concerned with the coverage of the senses of each word. Therefore, Wordnet contains as many senses as possible for each word. For example, the word "computer" is not ambiguous in computer science (for the collection CACM). However, two senses have been stored for it. For other words such as

"system" there are even more senses. In such a situation, we cannot use synonymy relations to expand queries with a strong confidence. In the expanded queries, a lot of "synonyms" added will be irrelevant to the application area, thus lead to noise documents. This is why the synonymy relations in Wordnet have very low strength after revisions.

On the other hand, the number of meronymy relations in Wordnet is much lower. When a meronymy relation is stored in Wordnet, there is usually a very strong relationship between the terms. Therefore, we can use them to expand queries with quite high confidence.

Comparison of the system's performance

We tested the approach with two different triangular norms: *min* and multiplication of real numbers. Our experiments showed that the second triangular norm gives much better results than the first in every case. Here, we only report the results with the multiplication of real numbers.

The system's performance is measured in terms of *average precision* over 11 recall points (0.0, 0.1, 0.2, …, 1.0). The following table gives a comparison of the system's performance using the baseline approach (Boolean query evaluation), our inferential approach before learning (i.e. with relevance strength for any relation = 1), and after learning for each type of relation. In the inferential approach, the query expansion process has been applied in lengths 1, 2 and 3 respectively.

Infer. Length	Approach	test 1	test 2	test 3	test 4	test 5	Average precision	increase (%)
	Baseline	23.81	14.76	21.93	19.81	14.77	19.02	*
L=1	Initial strength	25.57	19.08	20.86	17.65	16.27	19.89	4.61
	After learning	28.22	22.99	26.46	20.31	17.58	23.11	21.58
L=2	Initial strength	24.26	15.64	13.01	16.31	13.91	16.63	-12.54
	After learning	28.85	25.51	27.80	26.26	18.81	25.45	33.86
L=3	Initial strength	19.86	10.02	8.27	13.08	12.69	12.78	-32.75
	After learning	29.39	25.64	28.30	27.43	19.73	26.10	37.29

Table 2. Comparison of the system performance

(from (Nie and Brisebois 1996)).

We can see that with a coarse utilization of the thesaurus (the initial strength = 1), the more the thesaurus is used in inference process (the higher L is), the worse the system performs. However, with a reasonable assignment of relevance strengths (after learning), the more it is used in the inference process, the better the system performs. This observation may give an explanation to the negative impact of the same thesaurus to the system performance found in some previous experiments (Voorhees 1993, Voorhees 1994) .

Learning for individual relations

In order to compare the learning for different types of relations with the learning for individual relations, after the first learning, we used the same training data to adjust the relevance strengths of individual relations. In some cases, this individual adjustment succeeds in finding better relevance strength for relations. Here are some examples.

computer $\rightarrow_{0.27}$ data processor, electronic computer, information processing system

$\rightarrow_{0.0045}$ calculator, reckoner, figurer, estimator

file $\rightarrow_{0.094}$ data file

$\rightarrow_{0.0113}$ single file, Indian file

$\rightarrow_{0.0082}$ file cabinet, filing cabinet

In these examples, appropriate synsets (in computer science) are attributed with higher relevance strength. However, this adjustment fails in some other cases. For example,

hardware $\rightarrow_{0.143}$ hardware (artifacts made of metal)

$\rightarrow_{0.041}$ hardware (major items of military equipment)

$\rightarrow_{0.0129}$ hardware, computer hardware

As a consequence, the global system performance after individual adjustment only increased marginally. The average precision is changed from 26.10% to 27.04% for inference length 3. The minor difference may be explained by the lack of sufficient training data. A set of 40 queries is not enough for revision appropriately a number of individual relations. However, we think the process may be used in practice where more relevance feedback may be provided by the user. In comparison with the usual utilization of relevance feedback to reformulate one query, our utilization aims to form a knowledge base tailored to the user. Relevance feedback has a much longer effect on the user's queries. In such a new context, it is easier to obtain relevance

feedback from the user.

7. Cross-language IR - another application of the model

The inferential approach described in section 4 is not only applicable to monolingual IR. It may also be used in cross-language IR (CLIR), i.e. to retrieve documents written in one language with a query written in another language. In fact, the key problem in CLIR is to obtain an appropriate translation of the query. One of the ways to do this is through using of a bilingual dictionary. In fact, a bilingual dictionary may be considered as a particular form of thesaurus: a translation is a synonym of a term in certain context. As in the general inferential approach, we also have to select the appropriate terms to put into the new query, because words usually have various translations. Again, a similar approach of knowledge revision according to relevance feedback may be used to adapt a bilingual dictionary to a particular application context.

Yet another approach to CLIR is to use parallel texts to train a probabilistic model. This model determines the probability of each translation of a word. The translation relationship may be viewed as an uncertain logical implication. If b is a translation of a with the probability β, then we can consider that $a \rightarrow_\beta b$ is a piece of the system knowledge. The translation of a query $q = a$ is $q' = b^\beta$.

Recently, we applied this approach to CLIR between French and English using the TREC data (Voorhees and Harman 1997). A probabilistic model is first trained using a parallel corpus HANSARD which contains a set of English texts (Canadian parliament debates) and their translations in French (or vise versa). From this corpus, we estimate the probability $P(e|f)$ and $P(f|e)$ for any French word f and English word e. These translation relationships may be considered as the following logical implications:

$f \rightarrow_{P(e|f)} e$ and $e \rightarrow_{P(f|e)} f$

The principle of estimating $P(f|e)$ and $P(e|f)$ are as follows: the more two words e and f appear in the corresponding English and French sentences, the higher is the probabilities of their translation from one to another. We do not give details of this estimation here. Interested readers may refer to (Brown et al. 1992) and (Nie et al. 1999).

We used vector space model in this test. For each word in the original query, all the translations, together with their translation probability, is put into the translation vector. As the translations are sparse (due to the sparse-data problem), the number of translations of a word is big, but many of them are not related to the original word. Therefore, we only keep the top n translation words with the strongest probabilities as the translation of an original query. In our experiments, the best performances are obtained with the values of n varying between 15 and 50. This approach has proven to be very effective: We obtained comparable effectiveness to that using one of the best machine translation systems (Systran).

This short description of CLIR application is to show that the proposed inferential IR approach is indeed very general. It may be applied in various contexts for different purposes. An important problem of modern IR is to try to overcome the word gap between document and query. The inferential approach provides a general solution to this problem. Yet it is similar to the traditional query expansion approach.

8. Concluding remarks

Classical IR relies on direct word comparison to estimate document relevance. Inference is becoming an important operation in modern IR. In this paper, we defined an inferential approach to IR within a logic framework. The approach is general, yet simple and natural. It can be compared with the traditional query expansion: the latter is a particular case of the general

approach.

The key element of the approach is a knowledge base that contains a set of uncertain logical implications. However, such knowledge base does not exist. Instead, we have several thesauri in which terms are connected with semantic relations. We proposed a method to adapt such a thesaurus to a knowledge base of the required form using relevance feedback. Our experiments showed that the approach is effective in practice. The scale of the experiments is still limited. We are testing the approach using bigger test corpora such as those provided by the TREC project (Voorhees and Harman 1997).

Relevance feedback has been used in IR to expand the user's queries. However, the effect is short-term: it only affects the query that is being treated. Once a new query is input, the whole process should restart. In our case, we use relevance feedback to revise the system knowledge. This produces long-term effects. We think that this new process is more reasonable in practice and can obtain necessary cooperation from the user.

References

A. Bookstein (1983). Outline of a general probabilistic retrieval model. *Journal of Documentation*, 39(2): 63-72.

P. F. Brown, S. A. D. Pietra, V. D. J. Pietra, and R. L. Mercer (1992). The mathematics of machine translation: Parameter estimation. *Computational Linguistics*. 19: 263-312.

P.D. Bruza and T.W.C Huibers (1994). Investigating aboutness axioms using information fields, *Research and Development on Information Retrieval - ACM-SIGIR*, Dublin, pp. 112-121.

D. A. Buell (1982). An analysis of some fuzzy subset: applications to information retrieval systems. *Fuzzy Sets and Systems*, 7: 35-42.

Crestani and C.J. van Rijsbergen (1995). Information retrieval by logical imaging, *Journal of Documentation*, 51(1): 3-17.

W. B. Croft (1987). Approaches to intelligent information retrieval. *Information Processing & Management*, 23(4): 249-254.

D. Dubois and H. Prade (1984). Fuzzy logics and the generalized modus ponens revisited. *Cybernetics and Systems: An International Journal*, 15: 293-331.

J. Fagin (1988). *Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and Non-Syntactic Methods*, Ph.D. thesis, Department of Computer Science, Cornell University.

G. Grefenstette (1992). Use of syntactic context to produce term association lists. *Research and Development on Information Retrieval - ACM-SIGIR*, 89-97.

M. A. Hearst (1992). Automatic acquisition of hyponyms from large text corpora. *Fourteenth International Conference on Computational Linguistics COLING'92*.

D. Hindle (1989). Acquiring disambiguation rules from text. *27th Annual Meeting of the Association for Computational Linguistics*, 118-125, Pittsburgh.

M. Lalmas (1996). *Theory of Information and Uncertainty for the Modeling of Information Retrieval: An Application of Situation Theory and Dempster-Shafer's Theory of Evidence.* Ph.D. thesis, University of Glasgow.

M. Maron and J. Kuhns (1960). On relevance, probabilistic indexing and information retrieval. *Journal of the ACM*, 7: 216-244.

G. Miller (ed.) (1990). *Wordnet: an on-line lexical database, International Journal of Lexicography.*

J.-Y. Nie, M. Brisebois (1996). An Inferential Approach to Information Retrieval and its Implementation using a Manual Thesaurus, *Artificial Intelligence Review*, 10: 409-439.

J.-Y. Nie, M. Brisebois and F. Lepage (1996). Information retrieval as counterfactual, *The Computer Journal*, 38(8): 643-657.

J.-Y. Nie, M. Simard, P. Isabelle and R. Durand (1999). Cross-Language Information Retrieval based on Parallel Texts and Automatic Mining of Parallel Texts in the Web, *Research and Development on Information Retrieval - ACM-SIGIR*, Berkeley, pp. 74-81.

J. Pearl (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann: San Mateo CA.

H. J. Peat and P. Willett (1991). The limitation of term co-occurrence data for query expansion in document retrieval systems. *Journal of the American Society for Information Science*, 42(5): 378-383.

M.F. Porter (1980). An algorithm for suffix stripping. *Program*, 14(3): 130-137.

Y. Qiu and H. P. Frei (1993). Concept based query expansion. *Research and Development in Information Retrieval, ACM-SIGIR*, 160-169.

R. Rada, J. Barlow, J. Potharst, P. Zanstra, and D. Bijstra (1991). Document ranking using an enriched thesaurus. *Journal of Documentation*, 47: 240-253.

T. Radecki (1979). Fuzzy set theoretical approach to document retrieval. *Information Processing & Management*, 15: 247-259.

C. J. van Rijsbergen (1977). A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, 33: 106-119.

C. J. van Rijsbergen (1979). *Information Retrieval*, 2nd ed. Butterworths: London.

C. J. van Rijsbergen (1986). A non-classical logic for information retrieval. *The Computer Journal*, 29(6): 481-485.

C. J. van Rijsbergen (1989). Towards an information logic. *Research and Development on Information Retrieval - ACM-SIGIR*, 77-86.

S. Robertson, M. Maron, and W. Cooper (1982). Probability of relevance: a unification of two competing models for document retrieval. *Information Technology: Research and Development*, 1: 1-21.

G. Salton and C. Buckley (1988). On the use of spreading activation methods in automatic information retrieval. *11th ACM-SIGIR Conference*. pp. 147-160.

G. Salton and M. J. McGill (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill.

J. Savoy (1993). Stemming of French words based on grammatical categories. Journal of the American Society for Information Science, 44(1): 1-9.

J. Sinclair (1991). *Corpus, concordance, collocation*. Oxford University Press: Oxford.

K. Sparck-Jones (1991). Notes and references on early automatic classification work. *SIGIR Forum*, 25(1): 10-17.

H. Turtle and W. B. Croft (1990). Inference network for document retrieval. *Research and Development on Information Retrieval - ACM-SIGIR*, Brussels, pp. 1-24.

E. M. Voorhees (1993). Using Wordnet to disambiguate word senses for text retrieval. *Research and Development on Information Retrieval - ACM-SIGIR*, Pittsburgh, pp. 171-180.

E. M. Voorhees (1994). Query expansion using lexical-semantic relations. *Research and Development on Information Retrieval - ACM-SIGIR*, Dublin, 61-70.

E. M. Voorhees and D. K. Harman (eds.) (1997), Text REtrieval Conference (TREC-6). Gaithersburg, NIST SP 500-240.

W. G. Waller and D. H. Kraft (1979). A mathematical model for a weighted Boolean retrieval system. *Information Processing & Management*, 15: 235-245.