The use of relation matching in information retrieval

Khoo, Christopher S. G.

1997

Khoo, C. S. G. (1997). The use of relation matching in information retrieval. Library and Information Science Research E-Journal, 7(2), 1-20.
https://dx.doi.org/10.32655/LIBRES.1997.2.1

https://hdl.handle.net/10356/152355

https://doi.org/10.32655/LIBRES.1997.2.1

LIBRES: Library and Information Science Research
Electronic Journal ISSN 1058-6768
1997 Volume 7 Issue 2; September 30.
Bi-Annual LIBRE7N2 KHOO
DOI: https://doi.org/10.32655/LIBRES.1997.2.1

The Use of Relation Matching in Information Retrieval

Christopher Soo-Guan Khoo

Division of Information Studies, School of Applied Science,
Nanyang Technological University
assgkhoo@ntu.edu.sg

Abstract. Most information retrieval systems use keyword-matching to identify documents that may be relevant to the user's query. The user's query may contain multiple concepts linked together with relations. However, keyword matching methods ignore relations that are expressed in the query. This paper discusses the use of relation matching in information retrieval – matching terms and relations between terms expressed in the query with terms and relations found in the documents. Previous studies on relation matching are surveyed, and related issues and problems are discussed. The results of research carried out as part of the author's dissertation are reported and current research efforts are outlined.

Introduction

Information retrieval (IR) systems deal mainly with textual records written in a particular language. The records may be full-text documents, abstracts or just titles with bibliographic information. Even multimedia databases contain a substantial amount of text. Natural language text expresses concepts and the relations that hold between the concepts. It is therefore surprising that the retrieval methods used in most IR systems focus on keywords or concepts, and largely ignore the relations between concepts.

Researchers and system designers have found it difficult to make effective use of relations in IR, and have therefore chosen to devote their energies to refining keyword matching algorithms. There was widespread feeling among IR researchers in the late 1980s and early 1990s that the maximum attainable retrieval performance using keyword matching methods had nearly been reached and that substantial retrieval improvement was not expected using

these methods (Salton, Buckley & Smith; Smeaton, 1990). In the past few years, the growth of full-text databases and the World Wide Web gave research in keyword matching methods a new lease of life. Recent research has focused on adapting and refining keyword matching algorithms for full-text documents. Despite a large body of research on keyword matching, the effectiveness of retrieval systems is still rather low (Sparck Jones, 1997). Given the disappointing performance of keyword matching, other approaches to IR should continue to be actively explored. I feel that the use of relations in IR is a promising research area that can yield a breakthrough in the design of more effective IR systems.

There are two ways in which relations can improve information retrieval effectiveness – through *relation matching* and through *query expansion* with related terms.

In relation matching, both the concepts and the *relations* between the concepts as expressed in the user's query are matched with concepts and relations in documents. The system first performs concept or word matching, and for the query concepts that are found in the document, the system further checks whether the relations expressed between the query concepts match the relations expressed between the document concepts.

Consider the sentence *Harry loves Sally*. In the sentence, Harry is the *experiencer* of amorous feelings -- i.e. the *experiencer* relation holds between the concepts *Harry* and *love*. Sally is the beloved – i.e. the *patient* or *object* relation holds between *Sally* and *love*. If we use the sentence as a query in a keyword matching system, the system would look for documents containing the terms *Harry*, Sally** and *love** (where "*" is the truncation sign), and would not be able to distinguish among the following sentences:

(1) Harry loves Sally.
(2) Sally loves Harry, but Harry hates Sally.
(3) Harry's best friend loves Sally's best friend.
(4) Harry and Sally loves pizza.
(5) Harry's love for Sally is beyond doubt.

In fact, the words *Harry*, *Sally*, and *love* may occur so far apart in the document that they may not bear any relation to one another. With relation matching, a document that not only has the keywords *Harry*, *Sally* and *love* but also expresses the correct relation between the concepts would be given a higher ranking in the retrieval results (assuming the system produces ranked output) than a document which contains the keywords but not the correct relations. For the example sentences above, sentences (1) and (5) should be ranked higher than the other sentences.

The example above is not a realistic query, but it illustrates a problem with keyword matching. We cannot assume that the desired relations are expressed in a document solely from the fact that the words or concepts occur somewhere in the document or even within the same sentence. It seems reasonable to expect an IR system that performs relation matching to be more effective than one that performs only keyword matching. Relation matching improves *retrieval precision* by reducing the number of non-relevant documents retrieved (or reducing the ranking of such documents). The system uses the additional criteria of relation matches to eliminate some non-relevant documents that would otherwise

be retrieved by keyword matching.

A radically different way of using relations in IR is to expand query words with additional related words. Query expansion is used to improve *recall* in IR, i.e. to increase the number of relevant documents retrieved. The retrieval system will retrieve not only the documents that contain the query words but also documents containing words that are *related* to the query words. The synonym relation is most often used to expand query words, but other types of relations can also be used. As an example, suppose the user wants to find articles on the *eating habits of Singaporeans*. The term *eating* can be expanded using the following types of relations:

- synonym, e.g. *gobble, slurp, pig out*
- location, e.g. *restaurant*, *food court*, *hawker centre*
- instrument, e.g. *chopstick, bowl, plate*
- patient or object, e.g. *food*, *durian*, *Hokkien mee, buffet*
- manner, e.g. *voracious, messy*
- temporal, e.g. *breakfast*, *lunch*, *dinner*
- cause, e.g. *greedy, celebration, wedding.*

The two uses of relations in IR, relation matching and query expansion, reflect the distinction made by linguists between syntagmatic and paradigmatic relations. A *syntagmatic relation* between two words is the relation that is sythesized or expressed when a sentence containing the words are formed. The *Encyclopedia of Language and Linguistics* (Asher, 1994) defines it as "the relation a linguistic unit bears to other units with which it co-occurs in a sequence or context" (v.10, p.5178). A *paradigmatic relation*, on the other hand, is the relation between two words that is inherent in the meaning of the words. The *Encyclopedia of Language and Linguistics* defines it as the relation between linguistic units where one unit can substitute for the other according to different linguistic environments (v.10, p.5153). Relation matching deals with syntagmatic relations expressed in queries and documents. Query expansion makes use of paradigmatic relations to expand query words. However, the distinction between syntagmatic and paradigmatic relations is not clear cut. Afterall, paradigmatic relations can be expressed in sentences.

This paper focuses on relation matching. Query expansion with various types of relations is left to another paper. This paper examines the usefulness of relation matching in IR, analyzes the problems that have limited its usefulness and suggests promising directions for research. A review of the literature and a summary of the results of my dissertation research are presented. Although the use of manually identified relations is briefly reviewed, the focus is on relations that are automatically identified in text using computer algorithms and natural language processing techniques.

The main issues and research questions relating to relation matching are:

1. Comparison with keyword matching.

   To what extent does the use of relation matching improve retrieval effectiveness compared with keyword matching alone?

2. Comparison with word proximity matching (i.e. assuming that the desired relation occurs between the keywords just from the fact that they occur close together in the document. This issue is important because determining how far apart two keywords occur in the text is much easier and computationally much less expensive than identifying the relation between the words using natural language processing techniques. If proximity information gives results that are as good as relation matching, then there is no reason to perform automatic identification of relations and relation matching.

> To what extent does the use of relation matching improve retrieval results compared with using word proximity information?

3. The difficulty of identifying relations automatically, especially when the database is not limited to a narrow subject area.

> How can the automatic identification of relations in a heterogeneous textual database be improved?
>
> Are simple methods of identifying relations (and the relatively low accuracy) good enough to yield a material improvement in retrieval effectiveness?
>
> Will more accurate identification of relations yield better retrieval results than simple methods?

4. The relation matching method. There are several ways of identifying relations automatically and several ways of performing relation matching. There are also different types of relations and different sets of relations used by different researchers. All these can affect retrieval effectiveness.

5. The method of combining relation with keyword matching.

> What is the relative importance of keyword and relation matches in information retrieval?
>
> How can relation matching be combined with keyword matching to estimate the likelihood that a document is relevant to the user?
>
> Should different types of relations be weighted differently?

6. The circumstances in which relation matching is important. Relation matching may not be helpful in all situations.

> For what types of queries, documents, subject areas and applications is relation matching helpful in improving retrieval results?
>
> How can a retrieval system be designed to identify the instances when relation matching is likely to improve the retrieval results?

The first two issues, comparison with keyword matching and with word-proximity matching, deal with the basic research questions of whether relation matching really helps in IR. The other four issues deal with factors that possibly affect the usefulness of relation matching: what is the best way to use relation matching, and in what circumstances is relation matching useful. It is clear that there is much research work to be done in the area of relation matching. The various factors listed above interact, making it difficult to find the best combination of factors or methods to use for relation matching.

I shall first discuss briefly the use of *manually* identified relations and then review the work that has been done with *automatically* identified relations.

Matching With Manually Identified Relations

In some bibliographic databases, documents are indexed using manually assigned indexing terms that are "precoordinated", i.e. some human indexer has indicated that there is a relationship between two or more concepts in the content of the document. The *type* of relation between the concepts is usually not specified in precoordinated indexing but is implied by the context. This is the case with faceted classification schemes like Precis[1] (Austin, 1984) and Ranganathan's Colon classification (Kishore, 1986; Ranganathan, 1965). Precoordinate indexing allows the user to specify that there is some kind of relation between two terms when searching the subject descriptor field of the database.

Farradane (1967) pointed out that the implied relations in precoordinate indexing are unambiguous only in a narrow domain. In a heterogenous database covering a wide subject area, the relation between the precoordinated terms may be ambiguous. Farradane (1967) criticized the use of *implicit* relations as being "either too vague and open to error in use, or interfere with the meaning of the concepts" (p. 298).

Two indexing systems that make *explicit* use of relations are Farradane's (1950, 1952 and 1967) relational classification system and the SYNTOL model (Gardin, 1965; Levy, 1967). Farradane (1967) used nine types of relations: concurrence, equivalence, distinctness, self-activity, dimensional, reaction, association, appurtenance, functional dependence. The SYNTOL project used four main types of relations that were subdivided into finer relations (Gardin, 1965):

- Coordinative (Formal)
- Consecutive (Dynamic)
- Predicative
- Associative (Intrinsic)

    - Active

        - Agent
        - Patient

    - Inactive

        - Qualification

        - Inclusion

    - Circumstantial

        - Location
        - Means
        - Goal
        - Sign

A discussion of the theoretical issues related to the use of syntagmatic relations in indexing languages can be found in Green (1995),

It is not known whether the use of explicit relations in indexing really improves retrieval effectiveness compared with using individual index terms or using implicit relations (as in precoordinate indexing). The Aberystwyth Index Languages Test (Keen, 1973) found only a small improvement in retrieval precision for a minority of queries (13%) using Farradane's relations compared with not using relations. I have not found any other study that has attempted to evaluate the effectiveness of relation indexing.

The use of relations in manual indexing has not caught on probably because its effectiveness has not been clearly demonstrated and the effective use of relations requires training both for the indexer and the user. Without understanding the theoretical basis for a particular set of relations, the relations can look rather mysterious. Indexers have enough trouble assigning indexing terms (thesaural terms) to documents without the problem of assigning relations as well. Moreover, for many bibliographic databases, the indexers assign subject headings to indicate the general topic of the document rather than to describe the intellectual content of the document. With this kind of topical indexing, relations are less relevant.

Matching With Automatically Identified Relations

Automatic identification of relations and relation matching are performed on the "free text" portion of database records, i.e. the title, abstract and full-text if available. The systems can make use of either *syntactic* relations or *semantic* relations.

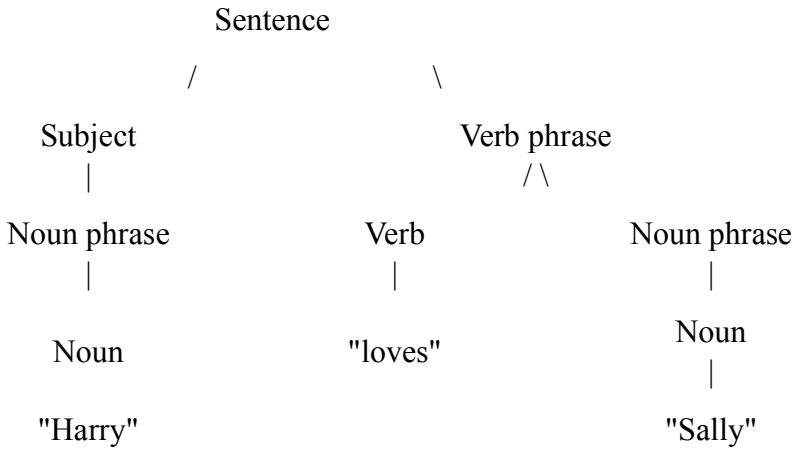
It is very difficult to identify semantic relations accurately using automatic methods. It requires complex computer processing and extensive general knowledge as well as domain-specific knowledge (i.e. subject knowledge) which often have to be coded by hand. Some researchers have therefore chosen to focus on syntactic relations as a substitute for semantic relations.

Use of Syntactic Relations

Syntactic relations refer to relations that are derived from the syntactic structure of the sentence. Determining the syntactic structure of a sentence is one of the processing steps needed for determining the semantic relations and the meaning of the sentence. So, by using syntactic relations instead of semantic relations, the amount of computer processing is reduced substantially. It should be noted that syntactic processing is itself a difficult

problem in natural language processing and accurate syntactic processing requires some semantic information!

There are different ways of representing the syntactic structure of a sentence. The following is a simple syntactic representation of the sentence *Harry loves Sally*:



This tree structure that represents the syntactic structure of the sentence is called a *parse tree*, and the process of producing this parse tree is called *parsing*. The parse tree indicates that the noun phrase *Harry* has the subject relation to the verb phrase *loves Sally*, and that the noun phrase *Sally* has the object relation to the verb *love*.

Examples of systems that use some form of syntactic parsing are the Constituent Object Parser (Metzler & Haas, 1989; Metzler, Haas, Cosic & Wheeler, 1989; Metzler, Haas, Cosic & Weise, 1990) and the TINA system (Schwarz, 1990a; Schwarz, 1990b; Ruge, Schwarz, & Warner, 1991). These systems perform syntactic processing on queries and documents to produce dependency trees that indicate which terms modify which other terms.

I noted earlier that syntactic parsing is difficult to perform accurately. Some researchers have used the simpler if less accurate method of *syntactic pattern matching* as a substitute for syntactic parsing. In syntactic pattern matching, the text is first tagged with part-of-speech labels and the system then identifies sequences of part-of-speech labels that match entries in a dictionary of acceptable patterns. An example of a syntactic pattern is *adjective* followed by *noun*. The researcher would construct a list of patterns that indicate relations that may be useful for information retrieval. Every phrase that matches a pattern in the dictionary is extracted and used as an index phrase. This is the method used by the FASIT system (Dillon & Gray, 1983) and the ALLOY system (Jones, deBessonet & Kundu, 1988)

After the syntactic relations in the query and the documents have been identified, the terms and relations in the documents have to be matched with those in the query. A retrieval score has to be calculated for each document to reflect the degree of match. There are two main

approaches to relation matching:

1. construct multi-word index phrases from the terms and relations, and match the index phrases constructed for the query with those for the documents.
2. match the syntactic tree structures produced from the query with those for the documents.

Relation matching using multi-word phrases that express the relation is simpler and more commonly used because the system can apply standard term matching techniques to the multi-word terms. The same procedure used for matching single-word terms is applied to the multi-word terms. If syntactic pattern matching is used to identify relations, as in the FASIT system (Dillon and Gray, 1983), then the output already consists of index phrases. The phrases have to be "normalized" so that different syntactic structures indicating the same relations are transformed into a "canonical form." For example, the phrase *retrieval of information* might be transformed into the phrase *information retrieval*.

If the output from the syntactic processing is some kind of tree representation of the syntactic structure, then the system can apply some rules to construct a set of index phrases from the syntactic tree. In the studies by Smeaton and van Rijsbergen (1988) and by Strzalkowski, Carballo and Marinescu (1995), pairs of terms with a dependency relation (i.e. where one term syntactically modified the other in the sentence) were extracted from the parse trees and used as multi-word index phrases.

The second method of relation matching, performing some kind of tree matching, is more complex but more flexible. Metzler and Haas (1989) and Sheridan and Smeaton (1992) pointed out that information is usually lost when a tree representation of a sentence structure is converted into index phrases. As an example, suppose that a document contains the noun phrase *patent information retrieval*. If two index phrases *patent information* and *information retrieval* are formed from the original noun phrase, then the document will not be retrieved if the query asks for *patent retrieval*. On the other hand, if the structure

patent -> information -> retrieval
(*patent* modifies *information* which modifies *retrieval*)

is preserved for matching. Then the system can take into account the "distance" in the relation between two terms to obtain a partial match with

patent -> retrieval

Breaking the original noun phrase structure into two index phrases also makes it impossible to distinguish between a document that contains the full phrase *the retrieval of patent information* and a document in which the phrases *patent information* and *information retrieval* appear separately but not together as one phrase, as in this example:

... retrieval of information about patent information offices.

Does syntactic relation matching yield a retrieval improvement compared with keyword

matching and word proximity matching? Research to date has found a small improvement when syntactic relations are taken into account in the retrieval process (Croft, 1986; Croft, Turtle & Lewis, 1991; Dillon & Gray, 1983; Smeaton & van Rijsbergen, 1988). The improvement over using just keywords is usually less than 10% in the *11-point recall-precision average* (one measure of retrieval effectiveness). Smeaton, O'Donnell and Kelledy (1995) obtained worse results from relation matching using a tree-matching procedure than from keyword matching.

Tree matching approaches appear to have fared worse then term matching techniques. This is probably because the optimal tree-matching technique for information retrieval is not yet known. Vector-based keyword matching methods have been studied for decades. Research on tree-matching methods for information retrieval has barely begun.

The retrieval performance from syntactic relation matching appears to be no better than and often worse than the performance obtainable using index phrases generated *using statistical methods* based on word proximity or co-occurrence in text, such as those described in Salton, Yang and Yu (1975) and Fagan (1989). Fagan (1989), who used a statistical non-syntactic procedure for constructing index, obtained retrieval improvements ranging from 2 to 23% depending on the document collection.

One possible reason why syntactic relation matching has not yielded better results than using word co-occurrence or word proximity information is the difficulty of identifying syntactic relations accurately. With the development of better parsers, bigger retrieval improvements may be obtained.

There are other ways in which past studies have been less than optimal in their use of syntactic relation matching. Croft, Turtle and Lewis (1991) performed syntactic processing on the queries but not on the documents. In one of their experiments, a phrase from the query was assumed to occur in the document if all the words in the phrase occurred somewhere in the document. In another experiment, the words in the query phrase were required to occur in close proximity in a document sentence. Similarly, Smeaton and van Rijsbergen (1988) performed syntactic processing on the queries but not on the documents. Furthermore, they processed only noun phrases.

Croft, Turtle and Lewis (1991) observed that the effect of using syntactic relations appeared to increase with the size of the database. The use of syntactic relations allows finer distinctions to be made between documents and this is likely to become more apparent the larger the document collection.

Use of Semantic Relations

Semantic relations, if identified accurately, can yield better retrieval results than *syntactic relations*. This is because a semantic relation can be expressed using different syntactic structures. A semantic relation is partly but not entirely determined by the syntactic structure of the sentence. For example, the same semantic relations between *Harry*, *love* and *Sally* as expressed in the sentence *Harry loves Sally* can be expressed in other ways:

- Harry’s love for Sally is beyond doubt.

- Harry declared his love for Sally.
- Sally is Harry’s one true love.
- Sally is the object of Harry’s love.
- Sally is the love of Harry’s life.
- Sally, Harry’s beloved, likes pizza.

Syntactic relation matching can fail to find a match between similar semantic relations if the relations are expressed using different syntactic structures.

IR systems that automatically identify semantic relations in text usually do it as part of the process of extracting information to store in a knowledge representation scheme or knowledge structure. Natural language text that has been converted to a knowledge representation is easier to manipulate by a computer. Information is retrieved from this knowledge store by comparing the information in the knowledge store with the knowledge representation of the user's query. Such a system is called a *conceptual information retrieval system*.

To extract information from text requires extensive domain-specific knowledge to support the syntactic and semantic processing. Since the domain knowledge often has to be hand-coded, such systems are usually limited to a narrow domain. In a specialized technical domain, sentence structures show less variety than in a non-technical or heterogeneous document collection. Moreover, the words used in a specialized technical domain may be limited to a relatively small technical vocabulary. Syntactic processing can thus focus on the syntactic structures commonly used in the subject area, and the construction of the domain knowledge as well as the semantic processing can focus on the terms and concepts that are important in that subject area.

Four examples of *conceptual information retrieval systems* are the RIME system (Berrut, 1990), the patent-claim retrieval system described by Nishida and Takamatsu (1982), the SCISOR system (Rau, 1987; Rau, Jacobs & Zernik, 1989) and the FERRET system (Mauldin, 1991). The RIME system was used for retrieving X-ray pictures associated with a medical report describing in natural language the content and medical interpretation of the picture. The system automatically converted the medical reports to binary tree structures that represented medical concepts and relations between them. The patent-claim retrieval system described by Nishida and Takamatsu (1982) extracted information from patent-claim sentences in patent documents and stored the information in a relational database. The SCISOR system extracted information from short newspaper stories in the domain of corporate takeovers, and stored the information in the "KODIAK" knowledge representation. The FERRET system has been used to convert astronomy texts to a frame representation.

It is not clear how effective these systems are. The system evaluations have not been carried out in a way that allows comparison with the best keyword matching methods.

Some researchers have studied particular types of semantic relations. Lu (1990) investigated the use of *case relation* matching using a small test database of abstracts. Case relations are the semantic relations that hold between a verb and the other constituents of a

sentence (Fillmore, 1968; Somers, 1987). In the example sentence *Harry loves Sally*, the case relation *experiencer* holds between *Harry* and *love*, and the case relation *patient* holds between *love* and *Sally*. The verb *love* is said to assign the case role of *experiencer* to the noun phrase *Harry* and the case role of *patient* to *Sally*. Using a tree-matching method for matching relations, Lu obtained worse results than from vector-based keyword matching. The tree-matching method used is probably not optimal for information retrieval and the results may not reflect the potential of relation matching.

Gay and Croft (1990) studied the semantic relations between members of compound nouns. A *compound noun* is a sequence of two or more nouns forming a unit that itself acts as a noun. Some examples are *college junior*, *junior college*, and *information retrieval*. The authors found that commonsense knowledge is essential for interpreting compound nouns. In a small experiment using the CACM document collection (Fox, 1983), the authors found that although their system correctly interpreted compound nouns about 76% of the time, it was not likely to yield a substantial improvement in retrieval effectiveness.

Identifying and coding the necessary domain knowledge for semantic processing is labor-intensive and time-consuming. Moreover, much of the knowledge is not portable to another domain. It is thus important to investigate whether non-domain specific procedures for extracting semantic relations can yield a material improvement in retrieval effectiveness.

The DR-LINK project (Liddy & Myaeng, 1993; Myaeng & Liddy, 1993; Myaeng, Khoo & Li, 1994) attempted to use general methods for extracting semantic relations for information retrieval. Non-domain specific resources like the *Longman Dictionary of Contemporary English* (2nd ed.) and *Roget's International Thesaurus* (3rd ed.) in machine-readable form were used. However, preliminary experiments found few relation matches between queries and documents.

Finally, Liu (1997) have investigated what I call *partial relation matching*. Instead of trying to match the whole concept-relation-concept triple (i.e. both concepts as well as the relation between them), he sought to match individual concepts together with the semantic role that the concept has in the sentence. In other words, instead of trying to find matches for "term1 ->(relation)-> term2", his system sought to find matches for "term1 ->(relation)" and "(relation)-> term2" separately. Liu used case roles and the vector-space retrieval model, and was able to obtain positive results only for long queries (abstracts that are used as queries).

Factors Affecting the Usefulness of Relation Matching and Implications for Research

Despite the generally disappointing results obtained in previous studies, I believe that there is still hope for relation matching. It is clearly not easy to make effective use of relation matching. To get positive results, the research has to be carried out very carefully. Positive results are more likely if we:

1. focus on high-level semantic relations
2. study one relation (or a small number of relations) at a time
3. extend current term-matching approaches to handle relations, rather than develop a new retrieval model.

This view is based on a consideration of the following factors that affect the usefulness of relation matching:

1. the accuracy of the automatic identification of relations
2. the method used for calculating the retrieval scores (e.g. a tree matching method or term matching method)
3. the type of documents and the type of queries.
4. the type of relations used and the set of relations used (i.e. syntactic or semantic, and which particular relations?)
5. the degree of *relational ambiguity* between the concepts linked by a relation.

I shall discuss these factors in greater detail.

Accuracy of automatic identification

It is difficult to identify relations in text accurately using automatic means. There is usually a substantial error rate using present day text processing computer programs. By focusing on one relation at a time or on a small number of relations, it is easier to increase the accuracy of the automatic relation identification. We can then get more authoritative results concerning the effectiveness of matching these relations (for queries containing these relations).

The retrieval method

We have seen that tree-matching methods have not worked as well as term-matching methods. Among the term-matching methods, the vector-based approach is the most well-studied (Salton, 1983). Using a vector-based approach one can easily compare the retrieval effectiveness of keyword matching versus keyword+relation matching using the same method for calculating retrieval scores.

Type of documents and type of queries

Relation matching may not work equally well for all queries and all types of documents. By studying a small number of relations and applying them to a range of queries and different document collections, we are more likely to be able to identify the circumstances when a particular type of relation is useful for improving retrieval effectiveness.

Type of relations

Semantic relations are preferable to syntactic relations because the same semantic relation can be expressed in many syntactic forms. In matching semantic relations, we are

performing matching across different syntactic relations. Syntactic relation matching may yield fewer matches than semantic relation matching. However, there are different types of semantic relations and some relations may be more useful than others for information retrieval. By focusing on one relation at a time, one can investigate the usefulness of each type of relation.

Relational ambiguity

The effectiveness of relation matching depends on what I call the *relational ambiguity factor*. Relation matching will give better retrieval results than term proximity matching to the extent that it is difficult to predict the relation between two terms just from the fact that they occur close together in a document. For example, if the words *eat* and *apple* occur in the same sentence, we can quite confidently predict that *apple* has the *patient* (i.e. object) relation to the word *eat* without even reading the sentence. There is little relational ambiguity in this case. If a query statement contains the relation "eat ->(patient)-> apple", using sophisticated natural language processing techniques to identify this relation will probably not yield better retrieval results than just searching "eat (same sentence) apple", i.e. searching for documents where *eat* and *apple* occur within the same sentence.

The greater the relational ambiguity between two query terms occurring in the document, the more likely will relation matching help improve retrieval. The degree of relational ambiguity between two terms increases with the distance between the terms in the text. If the words *eat* and *apple* are adjacent in the text, there is no doubt what the relation between the two terms is. However, if the two terms occur further apart, e.g. in adjacent sentences, then there is more doubt about what relation, if any, exists between the two terms. There is more relational ambiguity in this case.

Semantic relations can be at different "levels of abstraction." A high level relation is one that can be decomposed into more primitive concepts and relations. Consider the following semantic representation of the sentence *John eats an apple*:

John ->(eat)-> apple

The diagram expresses that John has an "eating" relationship with an apple. The relation *eat* is a high-level relation that can be decomposed into the concept *eat* and the "case relations" *agent* and *patient*:

John <-(agent)<- eat ->(patient)-> apple

Case relations are low-level semantic relations that exist between the main verb of a clause and the other constituents of the clause (Fillmore, 1968; Somers, 1987). Previous studies (e.g. Lu, 1990; Myaeng & Liddy, 1993) have not obtained good results with case relation matching. Case relations exist between terms that occur very close together in a sentence -- usually in adjacent positions and always within the same clause. With terms that occur so close together, relational ambiguity is less likely. Higher-level relations can exist between terms that occur further apart in the document – sometimes across sentences and

paragraphs. Relational ambiguity is more likely when terms are further apart. As mentioned earlier, relation matching is thus more likely to be helpful with higher-level relations than with case relations.

An In-Depth Study of One Relation

My Ph.D. dissertation research at Syracuse University (Khoo, 1995) was an in-depth study of just one relation – the cause-effect relation. An automatic method was developed for identifying cause-effect relations in text. The method did not use any domain-specific knowledge. It was successful in identifying and extracting about 68% of the cause-effect relations that were clearly expressed within a sentence or between adjacent sentences in full-text Wall Street Journal articles. Of the instances that the computer program identified as cause-effect relations, about 72% were correct.

The method was used in an experimental information retrieval system to identify cause-effect relations in a database of full-text *Wall Street Journal* documents. Causal relation matching was found to yield a small (2.9%) but significant improvement in retrieval results if the weights for causal relation matches were customized for each query using the user's relevance feedback on a sample of documents retrieved. The retrieval precision obtained was better at most of the "recall levels" along the recall-precision graph (Salton, 1983).

Causal relation matching did not yield better retrieval results than word proximity matching. However, combining causal relation matching with word proximity matching did produce better results than using word proximity matching alone, although the improvement was not significant. 24% of the 72 queries used obtained better retrieval results using causal relation matching plus word-proximity matching compared with using word proximity matching alone.

The small size of the retrieval improvement is not unexpected considering that the study make use only of one relation. There may be more than one relation in a query statement. As we incorporate more types of relation matching in the information retrieval strategy, more substantial improvements can be expected.

The retrieval improvement from causal relation matching was found to be greater for queries that obtained poor results from keyword matching than for queries that obtained good results from keyword matching. An analysis of a few queries using manually identified cause-effect relations indicated that bigger retrieval improvement could be expected with more accurate identification of causal relations.

Perhaps the most important insight obtained in the study was that partial relation matching where one member (i.e. term) of the relation is a wildcard is especially helpful. The most useful type of causal relation matching was found to be one where either the *cause* or the *effect* was not specified and could match with anything. This result suggests that if a query contains the relation "smoking causes cancer", then it is more useful to look for "smoking ->(cause)-> *" or "* ->(cause)-> cancer" (where "*" is a wildcard that can match with anything) than to look for the complete concept-relation-concept triple "smoking ->(cause)-> cancer". Substituting a wildcard for one member of the relation is effectively the same as assigning semantic roles or role operators to terms, as used by Liu (1997) and

Marega and Pazienza (1994).

Relation matching with a wildcard is helpful because it allows a match in the following cases:

1. when the document uses a synonym or related term that is not anticipated by the user. The wildcard allows the retrieval system to register a partial relation match when there is no match for one member of the relation.

2. when one member of the relation is specified in a different sentence in the document, as in the following two examples:

(1a) The policeman surprised a burglar.
(1b) In the ensuing struggle, he *killed* the burglar.

(2a) The policeman surprised a burglar.
(2b) The burglar *was killed* in the ensuing struggle.

In both examples, there is a causal connection between the policeman and the burglar's death. In example (1), an anaphor is used in the second sentence to refer to the policeman in the first sentence. In example (2), the policeman is not referred to in the second sentence but is implied by the context.

3. when one member of the relation is not specified in the query but is expected to be supplied by the relevant document, as in this example query:

I want documents that describe the consequences of the Gulf War.

This query can be represented as follows:

Gulf war ->(cause)-> *

The retrieval system will attempt to retrieve documents containing a relation that will "instantiate" the wildcard.

"Term1 ->(cause)-> term2" matching where both the cause and the effect have to find a match is less helpful because such matches are relatively rare. Word proximity matching should be used instead. Word proximity matching was found to give significantly better retrieval results than the baseline keyword matching method.

Future Work

In my dissertation work, I found that the weighting for causal relation matches had to be customized for individual queries. In other words, the importance of relation matching is different for different queries. In my study, a large number of relevance judgements were used to find out which types of relation matching were useful for each query and how each should be weighted. In a real life situation, it is not realistic to expect the user to give relevance judgements on a large sample of documents. It is thus important to develop a method for predicting, from a small number of relevance judgements or no relevance

judgements at all, which types of relation matching are likely to be useful for a particular query and what weighting to use.

My dissertation research focused on one particular semantic relation. Future work will investigate other semantic relations. The effect of one relation may be small and hard to detect. Processing two or more relations in a query may give us a clearer idea of the effectiveness of relation matching. It may also be that relations are more important for queries with multiple relations than for queries with one relation. Other researchers have found better results with long query statements. The effect of multiple relations may also be multiplicative rather than additive. Processing two relations in a query may yield more than twice the amount of improvement than handling one relation. This is a hypothesis to be investigated in the future.

---

1. In the Precis system, role operators are assigned to terms during the indexing process but are used only to determine the order of terms in the indexing string. Role operators are not explicitly used in the search process.

References

Asher, R.E. (Ed.). (1994). *The encyclopedia of language and linguistics*. Oxford: Pergamon Press.

Austin, D. (1984). *PRECIS: A manual of concept analysis and subject indexing.* (2nd ed.). London: British Library, Bibliographic Services Division.

Berrut, C. (1990). Indexing medical reports: the RIME approach. *Information Processing & Management, 26*(1), 93-109.

Croft, W. B. (1986). Boolean queries and term dependencies in probabilistic retrieval models. *Journal of the American Society for Information Science, 37*(2), 71-77.

Croft, W. B., Turtle, H. R., & Lewis, D. D. (1991). The Use of Phrases and Structured Queries in Information Retrieval. In A. Bookstein, Y. Chiaramella, G. Salton, & V.V. Raghavan (Eds.), *SIGIR '91: Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval* (pp. 32-45). New York: ACM Press.

Dillon, M., & Gray, A. S. (1983). FASIT: A fully automatic syntactically based indexing system. *Journal of the American Society for Information Science, 34*(2), 99-108.

Fagan, J. L. (1989). The effectiveness of a nonsyntactic approach to automatic

phrase indexing for document retrieval. *Journal of the American Society for Information Science, 40*(2), 115-132

Farradane, J.E.L. (1950). A scientific theory of classification and indexing and its practical applications. *Journal of Documentation, 6*(2), 83-99.

Farradane, J.E.L. (1952). A scientific theory of classification and indexing: further considerations. *Journal of Documentation, 8*(2), 73-92.

Farradane, J.E.L. (1967). Concept organization for information retrieval. *Information Storage and Retrieval, 3*(4), 297-314.

Fillmore, C. J. (1968). The case for case. In E. Bach & R. T. Harms (Eds.), *Universals in Linguistic Theory* (pp.1-88). New York : Holt, Rinehart and Winston.

Fox, E.A. (1983). *Characterization of two new experimental collections in computer and information science containing textual and bibliographic concepts* (Report No. TR83-561). Ithaca, NY: Department of Computer Science, Cornell University.

Gardin, J.-C. (1965). *SYNTOL.* New Brunswick, NJ: Graduate School of Library Service, Rutgers, The State University.

Gay, L.S., & Croft, W.B. (1990). Interpreting nominal compounds for information retrieval. *Information Processing & Management, 26*(1), 21-38.

Green, R. (1995). Syntagmatic relationships in index languages: A reassessment. *Library Quarterly, 65*(4), 365-385.

Jones, L. P., deBessonet, C., & Kundu, S. (1988). ALLOY: An amalgamation of expert, linguistic and statistical indexing methods. In Y. Chiaramella (Ed.), *11th International Conference on Research & Development in Information Retrieval* (pp. 191-199). New York: ACM.

Keen, E. M. (1973). The Aberystwyth index languages test. *Journal of Documentation, 29*(1), 1-35.

Khoo, C. S.G. (1995). Automatic identification of causal relations in text and their use for improving precision in information retrieval (Doctoral dissertation, Syracuse University, 1995).

Kishore, J. (1986). *Colon Classification: Enumerated & expanded schedules along with theoretical formulations*. New Delhi: Ess Ess Publications.

Levy, F. (1967). On the relative nature of relational factors in classifications. *Information Storage & Retrieval*, **3**(4), 315-329.

Liddy, E. D., & Myaeng, S. H. (1993). DR-LINK's linguistic-conceptual

approach to document detection. In D.K. Harman (Ed.), *The First Text REtrieval Conference (TREC-1)* (NIST Special Publication 500-207, pp. 1-20). Gaithersburg, MD: National Institute of Standards and Technology.

Liu, G.Z. (1997). Semantic vector space model: Implementation and evaluation. *Journal of the American Society for Information Science, 48*(5), 395-417.

Longman dictionary of contemporary English. (1987). 2nd ed. Harlow, Essex: Longman.

Lu, X. (1990). An application of case relations to document retrieval (Doctoral dissertation, University of Western Ontario, 1990). *Dissertation Abstracts International*, **52-10**, 3464A.

Marega, R., and Pazienza, M.T. (1994). CoDHIR: An information retrieval system based on semantic document representation. *Journal of Information Science, 20*(6), 399-412.

Mauldin, M.L. (1991). Retrieval performance in FERRET: A conceptual information retrieval system. In A. Bookstein, Y. Chiaramella, G. Salton, & V.V. Raghavan (Eds.), *SIGIR '91: Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval* (pp. 347-355). New York: ACM Press.

Metzler, D. P., & Haas, S. W. (1989). The Constituent Object Parser: Syntactic structure matching for information retrieval. *ACM Transactions on Information Systems*, **7**(3), 292-316.

Metzler, D. P., Haas, S. W., Cosic, C. L., & Weise, C. A. (1990). Conjunction, ellipsis, and other discontinuous constituents in the constituent object parser. *Information Processing & Management, 26*(1), 53-71.

Metzler, D. P., Haas, S. W., Cosic, C. L., & Wheeler, L.H. (1989). Constituent object parsing for information retrieval and similar text processing problems. *Journal of the American Society for Information Science, 40*(6), 398-423.

Myaeng, S. H., & Liddy, E.D. (1993). Information retrieval with semantic representation of texts. In *Proceedings of the 2nd Annual Symposium on Document Analysis and Information Retrieval* (pp. 201-215).

Myaeng, S. H., Khoo, C., & Li, M. (1994). Linguistic processing of text for a large-scale conceptual information retrieval system. In Tepfenhart, W.M., Dick, J.P., & Sowa, J.F. (Eds.), *Conceptual Structures: Current Practices: Second International Conference on Conceptual Structures, ICCS '94* (pp. 69-83). Berlin: Springer-Verlag.

Nishida, F., & Takamatsu, S. (1982). Structured-information extraction from

patent-claim sentences. *Information Processing & Management, 18*(1), 1-13.

Ranganathan, S.R. (1965). *The Colon Classification*. New Brunswick, N.J.: Graduate School of Library Service, Rutgers, the State University.

Rau, L. (1987). Knowledge organization and access in a conceptual information system. *Information Processing & Management, 23*(4), 269-283.

Rau, L.F., Jacobs, P. S., & Zernik, U. (1989). Information extraction and text summarization using linguistic knowledge acquisition. *Information Processing & Management, 25*(4), 419-428.

Roget's international thesaurus. (1962). 3rd ed. New York: Thomas Y. Crowell Company.

Ruge, G., Schwarz, C., & Warner, A.J. (1991). Effectiveness and efficiency in natural language processing for large amounts of text. *Journal of the American Society for Information Science, 42*(6), 450-456.

Salton, G., & McGill, M.J. (1983). *Introduction to modern information retrieval.* New York: McGraw-Hill.

Salton, G., Buckley, C., & Smith, M. (1990). On the application of syntactic methodologies in automatic text analysis. *Information Processing & Management, 26*(1), 73-92.

Salton, G., Yang, C.S., & Yu, C.T. (1975). A theory of term importance in automatic text analysis. *Journal of the American Society for Information Science, 26*(1), 33-44.

Schwarz, C. (1990a). Automatic syntactic analysis of free text. *Journal of the American Society for Information Science, 41*(6), 408-417.

Schwarz, C. (1990b). Content based text handling. *Information Processing & Management, 26*(2), 219-226.

Sheridan, P., & Smeaton, A.F. (1992). The application of morpho-syntactic language processing to effective phrase matching. *Information Processing & Management, 28*(3), 349-369.

Smeaton, A.F. (1990). Natural language processing and information retrieval. *Information Processing & Management, 26*(1), 19-20.

Smeaton, A.F., & van Rijsbergen, C.J. (1988). Experiments on incorporating syntactic processing of user queries into a document retrieval strategy. In Y. Chiaramella (Ed.), *11th International Conference on Research & Development in Information Retrieval* (pp. 31-51). New York: ACM.

Smeaton, A.F., O'Donnell, R., & Kelledy, F. (1995). Indexing structures

derived from syntax in TREC-3: System description. In D.K. Harman (Ed.), *Overview of the Third Text REtrieval Conference (TREC-3)* (NIST Special Publication 500-225, pp. 55-67). Gaithersburg, MD: National Institute of Standards and Technology.

Somers, H.L. (1987). *Valency and case in computational linguistics*. Edinburgh : Edinburgh University Press.

Sparck Jones, K. (1997, Feb. 10). Summary performance comparisons TREC-2, TREC-3, TREC-4, TREC-5 [Postscript file]. In *TREC-5 Proceedings*. Available: http://www-nlpir.nist.gov/TREC/trec5.papers /sparckjones.ps (visited 3 July 1997).

Strzalkowski, T., Carballo, J.P., & Marinescu, M. (1995). Natural language information retrieval: TREC-3 report. In D.K. Harman (Ed.), *Overview of the Third Text REtrieval Conference (TREC-3)* (NIST Special Publication 500-225, pp. 39-53). Gaithersburg, MD: National Institute of Standards and Technology.

This article was originally published in
_LIBRES: Library and Information Science
Electronic Journal_ (ISSN 1058-6768) September 30, 1997
Volume 7 Issue 2.

Christopher Soo-Guan Khoo
assgkhoo@ntu.edu.sg

Return to Libre7n2 Contents
Return to Libres Home Page