Las Vegas

Information Retrieval with Semantic Representation of Texts

Sung H. Myaeng & Elizabeth D. Liddy
School of Information Studies
Syracuse University
shmyaeng@mailbox.syr.edu; liddy@mailbox.syr.edu

ABSTRACT

As part of the DARPA TIPSTER program, we have developed an information retrieval system that is conceptually and linguistically oriented in its text processing and representation. Texts are processed using knowledge bases and represented using Conceptual Graphs, Subject Field Codes, and discours-level text structures. The rich representations are used at various stages of the system operation and the final output is generated by Conceptual Graph matching. We have tested our full system using a large collection, and the results are very encouraging. We also describe our continues effort to refine and better integrate modules and develop incomplete knowledge bases for better performance.

1. Introduction

DR-LINK (Document Retrieval using LINguistic Knowledge) is a conceptually and linguistically oriented information retrieval system that attempts to satisfy two seemingly conflicting requirements: the need to handle a vast amount of information; and the need to capture and process user's sophisticated information needs as well as the semantics of texts. The system employs a variety of linguistically-oriented techniques at different stages to process texts in a domain-independent manner and represent them at a conceptual level so that specific semantic constraints included in the user's information need can be dealt with.

The retrieval task for which our system has been designed and developed is somewhat different from that of traditional information retrieval systems. Instead of a set of carefully chosen keywords that are often assumed to be sufficient to represent the topicality of the relevant documents, the topic statements our system needs to process consist of natural language sentences that contain not only the general areas of interest, expressed as concepts, but also some specific semantic constraints that certain concepts must meet (e.g. the location or nationality or an action instigated by an entity). It is obvious that such constraints that are critical for relevance judgments cannot be met easily without considering text representations richer than keyword based representation. Our system produces and utilizes several different representations for different purposes, including the semantic representation based on Subject Field Code in Longman's Dictionary of Contemporary English (LDOCE) and text structure based representation. We use Conceptual Graphs (CG)[1] as the final representation with which documents are retrieved based on a uncertainty-theoretic CG matching algorithm.

In order to build the rich representation, we employ a variety of NLP techniques. Instead of relying on syntactic parsing, we have developed several special handlers that process different constituents in sentences, which belong to different grammatical categories as well as the knowledge bases necessary for the handlers to work. The current implementation includes verb case frame handler, preposition handler, and noun phrase handler, each of which attempt to provide a relation between two concepts. For example, the case frame handler's job is to consult the knowledge base (case frames) and connect the verb to the neighboring constituents (e.g. subject and object) with a relation. Other handlers like apposition handler, complex nominal handler, and adverbial handler will be included. We have been developing the knowledge bases primarily based on general-purpose, machine-readable resources (e.g. LDOCE), as well as corpus analysis, so that we can minimize our system's dependency on a particular domain model.

Another unique task characteristic we have considered for this project is related to the size of the databases to be processed. Compared to most of other previous IR research where experimental work has been limited to relatively small test collections, our effort has been to make our system capable of processing a large volume of data and testing it against large collections. This goal is particularly challenging since we employ a variety of sophisticated, computationally-expensive processes where other systems using NLP techniques and knowledge bases have been built for small databases or narrow domains. We handle this situation by filtering out a large number of irrelevant documents at an early stage so that fine-level matching can focus on a small subset of the entire database.

Our system is now fully functional but with minimal amount of knowledge in knowledge bases. Also some of the specialized handlers are yet to be included. Despite the incompleteness of the knowledge bases the system relies on and some of the system components, our preliminary results from the testing we did are very promising. The filtering part was tested with three different databases, Wall Street Journal, AP, and Ziff (1.5 GB together), and the full system with the Wall Street Journal (550MB) collection.

In the following, we first discuss the task characteristics in detail and describe how the CG representation of documents and topic statements are used in the full system to retrieve the final output of the documents. We then give a detailed description of how such representations are constructed from texts, together with the retrieval results. The next sections describes the techniques used in the other three modules of the system that create rich semantic representations of texts and their performance.

2. Topic Statements

In our task environment, the input to the system is a set of topic statements written by the users, each of which consists of several fields as in Fig.1. While important concepts (or keywords) are embedded in the natural language sentences under the <Description> and <Narrative> fields, the <Concepts> fields contain some additional concepts, some of which may occur in the natural language sentences. A unique aspect of the topic statements is that the natural language sentences often specify a variety of semantic constraints that cannot be expressed using a set of words or simple phrases alone. For example, a document matched with a list of words, (*current, agreement*, ...) doesn't necessarily mean that it is about *current*, as opposed to *past*, agreement. A list of keywords is not just powerful enough to represent the restriction that a *debtor* has to be a *developing country.*

The <Factors> field is sometimes included to explicitly specify some of the constraints. For example, the topic statement in Fig. 1 includes the fact that the *debt rescheduling agreement* has to be *current*. In order to help the person or the system that converts the topic statement into whatever representation to be used, definitions of some jargon are sometimes included although it is not entirely clear how

Topic: Debt Rescheduling

Description: Document will discuss a current debt rescheduling agreement between a developing country and one or more of its creditor(s).

Narrative:

A relevant document will discuss a current debt rescheduling agreement reached, proposed, or being negotiated between a debtor developing country and one or more of its creditors, commercial and/or official. It will identify the debtor country and the creditor(s), the repayment time period requested or granted, the monetary amount requested or covered by the accord, and the interest rate, proposed or set.

Concept(s):

1. rescheduling agreement, accord, settlement, pact
2. bank debt, commercial debt, foreign debt, trade debt, medium-term debt, long-term debt
3. negotiations, debt talks

...

...

Factor(s):

Nationality: Developing country
Time: Current

Definition(s):

Debt Rescheduling - Agreement between creditors and debtor to provide debt relief by altering the original payment terms of an existing debt. This is most often accomplished by ...

Figure 1: Topic Statement Example

they can be used automatically.

3. Conceptual Graph Representation

As a way to handle the semantic constraints, we opted for the Conceptual Graphs framework, which is a variation of semantic networks, as the underlying representation formalism for the final retrieval purpose. While the CG framework has many features (operators and knowledge structures) to offer for information retrieval [2], the current system uses only its basic network structure of concept and relation nodes, where a concept node can have a referent as well as the name. For example, a part of the topic statement in Fig. 1 can be represented as a CG in Fig. 2. where concept nodes in square brackets and relation nodes in regular parenthesis are in linear form. It should be noted that a concept node can be a proposition with a CG as a referent and that the question mark indicates that a referent is being sought in the topic statement (e.g. [creditor:*4 ? 0.3 0.0 0.7]).

While the original CGs don't have any stipulation for weights, we have extended the notation to include them only for the topic statement representation [3]. The first number on a node represents the degree of evidence that the existence of the node in a document CG will make it relevant to the topic statement. The second number on a node is for the degree of evidence that its existence will make the document irrelevant and becomes non-zero when the node is negated in the topic statement. The third number is obtained by subtracting the sum of the first two numbers from

```
[current 0.1 0.0 0.9] ->(STATUS 0.3 0.0 0.7)->
    [$proposition [agree 0.3 0.0 0.7] -
        (ACTIVITY 0.1 0.0 0.9)-> [developing_country 0.3 0.0 0.7]
        (AGENT 0.1 0.0 0.9)-> [creditor:*4 ? 0.3 0.0 0.7]
        (PATIENT 0.1 0.0 0.9)-> [reschedule:*3 0.3 0.0 0.7] -
                                        (PATIENT 0.1 0.0 0.9)-> [debt:*2 0.3 0.0 0.7],
        (LOCATION2 0.1 0.0 0.9)<- [reach 0.1 0.0 0.9]
        (PATIENT 0.1 0.0 0.9)<- [propose 0.1 0.0 0.9]
        (PATIENT 0.1 0.0 0.9)<- [negotiate 0.1 0.0 0.9] -
                    (CO-AGENT 0.1 0.0 0.9)-> [debtor_developing_country ? 0.2 0.0 0.8]
                    (CO-AGENT 0.1 0.0 0.9)-> [creditor:*4 ? 0.3 0.0 0.7] -
                              (CHAR 0.1 0.0 0.9)-> [commercial 0.1 0.0 0.9]
                              (CHAR 0.1 0.0 0.9)-> [official 0.1 0.0 0.9]
    ].
[repay 0.2 0.0 0.8] -
        (PATIENT 0.1 0.0 0.9)-> [debt 0.2 0.0 0.8]
        (TIME3 0.1 0.0 0.9)-> [duration 0.2 0.0 0.8] -
                  (PATIENT 0.1 0.0 0.9)<- [request 0.1 0.0 0.9]
                  (PATIENT 0.1 0.0 0.9)<- [grant 0.1 0.0 0.9].
[money 0.2 0.0 0.8] -
         (PATIENT 0.1 0.0 0.9)<- [request 0.1 0.0 0.9] -> (AGENT 0.1 0.0 0.9)-> [accord:*1 0.2 0.0 0.8],
         (PATIENT2 0.1 0.0 0.9)<- [cover 0.1 0.0 0.9]->(PATIENT1 0.1 0.0 0.9)-> [accord:*1 0.2 0.0 0.8].

[interest_rate 0.2 0.0 0.8] -
        (PATIENT 0.1 0.0 0.9)<- [propose 0.1 0.0 0.9]
        (PATIENT 0.1 0.0 0.9)<- [set 0.1 0.0 0.9].
```

Figure 2: CG Representation of a Topic Statement (Description and Narrative)

1 and interpreted as the uncertain portion of the certainty interval. A description of how the weights are used is in the next section.

The relations we are currently using for the CG representation were chosen with two design goals: 1) Since the system must function in a domain-independent manner, the relations must be domain-independent; and 2) since the text must be converted into CGs automatically, the relations must be extractable by an automatic text processing method. It became clear that in order to meet the goals, we would need to stay with linguistically oriented relations as much as possible so that they can be extracted by applying linguistic knowledge rather than domain-specific knowledge. Therefore, our choice of relations was based primarily on the linguistics literature (e.g. [4] and [5]). Appendix 1 shows the list of relations being used in the current system, which will evolve as we improve the knowledge bases and add in more capabilities in text processing.

4. Retrieval with CG Representation

When the topic statement and documents are represented in the CG form, the system computes the relevance of documents based on the CG matching algorithm [6] and the scoring scheme we have developed. The main function of the matching/scoring component is to determine the degree to which two CGs share a common structure. Given two CGs, one for the entire topic statement and the other for a unit of document (e.g. a sentence or a paragraph), the system first applies the matching algorithm to find all non-redundant, matching sub-graphs that we call a set of solutions. When all the solution sets or maximally joinable common sub-graphs are found for the entire document consisting of multiple CGs, the scoring algorithm is applied to compute the final score.

As a way to model plausible inferencing in information retrieval, we adopt both partial matching (e.g. between 'bank debt' and 'debt') and inexact matching (e.g. between two synonyms). While partial matching is primarily geared toward the handling of hyphenated words and proper noun phrases (e.g. 'Mr. John Adams' and 'Mr. Adams'), inexact matching allows for matching between two semantically close relations as well as between synonymous concepts. For this purpose, we have created a similarity table for relations, which lists pairs of matchable relation together with the similarity values. For instance, AGENT and CO-AGENT relations will match with the similarity value 0.8. Partial matching is also done with the '?' symbol that indicates the need to instantiate the concept node. For example, [company ?] in a topic statement CG can match with the same concept node with any referent (e.g. [company: IBM]).

The scoring scheme is based on a method to model information retrieval as plausible inferences with CGs [3]. Each solution (i.e. a matching sub-graph), regardless of its size, is considered as a piece of evidence that contributes to either relevance or irrelevance of the document with respect to the topic statement, depending on whether or not the sub-graph in the topic statement is negated. Individual pieces of evidence from the solution sets are then gathered to determine the relevance of the entire document. Dempster-Shafer's uncertainty combination rule [7] is used as the basic operation.

More specifically, given the topic statement CG in Fig. 2 where the numbers associated with nodes represent the *basic probability assignment* (weights) or *bpa*, the scoring algorithm "overlays" each of the solutions on top of it to see how much of the topic statement is covered by the document. The role of the weights is that regardless of how many layers have been laid on top of a topic statement CG node, the maximum score that can be contributed by the node toward the document relevance cannot exceed the weight. This restriction is a way of preventing frequently occurring concepts from dominating the overall score on the document. The two overlapping solutions in the example can come from either a single unit CG or multiple unit CGs in the document.

The score on each matched node (i.e. a node covered with at least one layer) is computed as an *orthogonal sum* [7] of the weights associated with the layers, each of which represent the degree of partial matching between the topic statement node and the document node. The sum is then normalized so that the maximum value is within the ceiling value. For C1, for example, the bpa values from the two matches are (.8, 0, .2) and (.5, 0, .5), respectively. The orthogonal sum of the two bpa's results in (.9, 0, .1) which then is normalized by the ceiling value (.3) to produce the final score (.27, 0, .73). The scores for all other nodes (R1, C2, R3, C4) are computed in the same way and combined with the orthogonal sum operation across the nodes to produce the final score for the entire matching.

This scoring scheme of "accumulating" scores on the matched topic statement nodes is a way to meet one of the expectations in the task environment. That is, a document would be relevant as long as it has a "hot spot" which is a piece of text that mentions some part of information included in the topic statement, regardless of the length or the major theme of the document. For example, a document whose

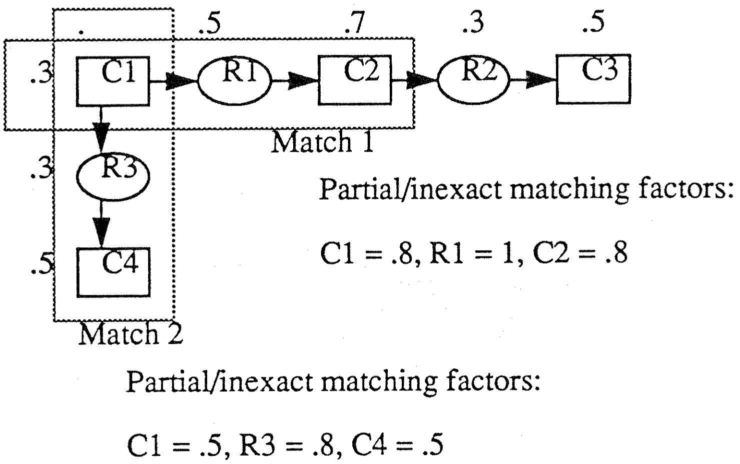
.
.5
.7
.3
.5
.3
C1
R1
C2
R2
C3
Match 1
.3
R3
Partial/inexact matching factors:
C1 = .8, R1 = 1, C2 = .8
.5
C4
Match 2
Partial/inexact matching factors:
C1 = .5, R3 = .8, C4 = .5

Figure 3: Example for Scoring Algorithm

major content is about *the problems with increasing medical insurance* would be relevant to the topic statement requiring information on *a breakthrough in medicine* if it has a sentence that mentions *a new drug on cancer therapy* as part of explaining a case study. In other words, the more portion of the topic statement is covered, the higher score the system gives to the document. With the partial/ inexact matching factors, however, frequency information is factored into the scheme to some extent. In the current implementation, the ceiling score on each node can be reached with a multiple exact matches.

One drawback of the scoring scheme described above is that it doesn't distinguish a case where a given area of the topic statement CG is covered by small multiple solutions from another case where it is covered by a single solution. In other words, it doesn't reward the document that has a solid, connected match as opposed to smaller matches scattered in the document. In order to take into account this difference, which amounts to a difference between keyword-based matching and structural matching to some extent, we normalize the score with a value estimating "connectivity" defined as:

$$\frac{\text{SUM (\#all nodes on a layer) / \#layers}}{\text{Total \# nodes in covered CG}}$$

In the above example, the final score becomes:

$$\frac{(3+3)/2}{5} = 3/5$$

5. Extracting Concepts and Relations

The building blocks for the CG representation are concept-relation-concept triples we generate automatically from texts. While concepts can be *identified* when different constituents in a sentence are marked appropriately (e.g. in terms of part-of-speech), relations needs to be *extracted* since they are not explicit in the sentence. The way we extract relations is to identify linguistic (lexical, syntactic, and semantic) patterns that reveal relations between concepts. These patterns are all captured in a form of rules [8] in knowledge bases that are consulted in the pattern identification process. In order to facilitate the concept and pattern identification processes, texts are first tagged with part-of-speech using the POST tagger [9] and bracketed for constituents boundaries (e.g. phrases and clauses).

In the Relation-Concept Detector (RCD) component, there are several sub-modules that process different parts of sentences, specializing in different grammatical categories with their own knowledge bases. They will be: verb case frame handler, apposition handler, complex nominal handler, preposition handler, nominalized verb handler, adverbial handler, and adhoc handler). It should be noted that these submodules as well as the associated knowledge bases are at different stages of development. For the current implementation, we focused on the implementation of the case frame handler [10] although some of the functions as well as its knowledge base are not complete. Some other sub-modules have been implemented in their skeletal form and others are yet to be included.

As an example, we show how the case frame handler detects concept-relation-concept triples (see [10] for details). It is activated whenever there is a verb in a certain form (e.g. a tensed verb or non-tensed verb like a gerund) in the text. The verb is looked up in the case frame knowledge base to retrieve a set of case frames that prescribe what constituents must exist with the verb and what relations must be assigned between each of the neighboring constituents and the verb. Given a sentence fragment:

> *... the spokesman declined to comment on the accident ...*

where the verb *decline* becomes the focus of attention, the following case frames will be retrieved:

```
(decline 1   (subject ? patient))
(decline 2   (subject human agent)
             (object1 ? patient))
(decline 3   (subject human agent)
             (to-verb ? activity))
```

where each constituent has three components: grammatical category, semantic restriction, and the relation to be assigned. The question mark is an indication that there is no known semantic restriction. The third frame is chosen based on the algorithm that tries to find the one that is instantiated best with the text being analyzed. As a result, the following triples are generated:

```
[decline:*1] -> (agent) -> [company]
[decline:*1] -> (activity) -> [elaborate]
```

Triples generated from the sub-modules are integrated to form a connected CG. At this stage, an inverted index for the CGs on concept names are constructed so that a subset of CGs can be selected efficiently when necessary. In the current implementation, CGs are constructed for individual sentences. Paragraph level or text-structure level CGs will be generated with a referent resolution method.

The full DR-LINK system was tested against the Wall Street Journal collection (550 MB) using 25 topic statements. While the topic statement CGs were constructed manually, no domain knowledge was brought in so that automatic processing of topic statements can be simulated. Document CGs were constructed

	DR-LINK	A	B	C	D
At 5 documents:	.68	.63	.29	.27	.15
At 10 documents:	.60	-	-	-	-
At 30 documents:	.55	.52	.23	.21	.14
At 100 documents:	.41	.39	.18	.19	.10
11-pt average:	.27	.36	.16	.10	.08

Table 1: Precision Values for DR-LINK and TREC Category B Systems

automatically. The output generated by the Subject Field Coder (SFC), Proper Noun (PN) Interpreter, and Text Structurer (TS), described below, was used as the input to test the RCD and the CG Matcher. Due to the time constraint, only the top 2000 documents in the ranked output from the SFC+PN+TS results were used for each topic statement. The top 500 documents from each run were sent to NIST for relevance judgments. The 11-point average precision for 22 topic statements for which we received the results in the first batch was 0.2697. Since only the top 100 documents for each run were included in the final pool of the documents judged for relevance, the precision values in Table 1 are good indicators for the effectiveness. A, B, C, and D represent the four "category B" systems in TREC, which were tested against the same Wall Street Journal collection using 25 topic statements. It should be noted, however, that the numbers are not directly comparable and thus used as rough indicators for how DR-LINK system performed. The results for the TREC systems are from 19 topic statements for routing as opposed to 22 topic statements for ad-hoc [11]. The 11-point average of the system A whose performance is similar to that of DR-LINK is .3583.

The recall value for top 500 judgments, averaged over the 22 topic statements, is .5027. However, this figure is misleading in the sense that many of the documents between ranks 101 and 500 were not included in the relevance judgment pool and simple considered irrelevant although some of them would be relevant. On the average, 296 documents in the top 500 documents were not evaluated in the relevance judgments.

Another factor that needs to be considered in interpreting the evaluation results is that the number of documents processed by the RCD and CG Matcher was limited to the top 2000 documents generated by SFC+PN+TS. Since not all relevant documents were included in the top 2000, which is an artifact imposed on the system due to the testing schedule and time constraints and will have to be increased for future testing, the performance of the RCD and CG Matcher was affected by the limit. When a greater number of documents from the filtering process are processed, both recall and precision are expected to improve. Considering the incompleteness of the knowledge bases and the RCD sub-modules, this level of performance is very encouraging.

6. System Orientation

Since this paper describes the DR-LINK System in the reverse order of it's actual processing of text, a re-orientation to the system view is necessitated at this point. As alluded to in the introduction, DR-LINK consists of six modules which, in combination, produce textual representations that capture great breadth and variety of semantic knowledge which will be used to improve retrieval effectiveness, in terms of both recall and precision. The full integration of the system will see the enrichments added to the text by the first three modules in the system exploited to their full advantage for the final system output of ranked documents. Lacking this level of integration when the system was formally tested for DARPA's eighteenth- month required evaluation, the first three system modules were evaluated on their individual output. In addition, the three modules were linked together and tested as a preliminary filter for producing a reasonable ranking of appropriate documents to be further processed by the Relation-Concept Detector and CG Generator/Matcher. It should be recognized, however, that this does not reflect their optimum performance in a fully integrated system.

While the Relation-Concept Detector described above processes individual grammatical categories such as verbs and complex nominals to extract semantic relations within the sentence for inclusion and matching in the CG formalism, the following modules produce semantically rich representations at other levels of linguistic analysis. In particular, the Subject Field Coder assigns semantic values to individual words which are disambiguated by context and then used to produce a text- level semantic summary; the Proper Noun Interpreter assigns very specific roles/relations at the conceptual category level, again disambiguated by context; and the Text-Structurer produces a discourse level organization and representation of document content which is unique to DR-LINK. Although many developers of text processing systems (particularly the MUC systems) have long advocated the need for discourse knowledge in their systems, the Text Structurer in the DR-LINK System is the first implementation of full discourse structuring of texts that we are aware of.

7. Subject Field Coder

The first module in the system, Subject Field Coder (SFCer), adds to the document a summary-level semantic representation of each text's contents that is usable both for prioritizing a large set of newly arriving documents for their broad subject appropriateness to a standing query, or for dividing a database into clusters of documents pertaining to the same subject area. The clustered database provides an intuitive organization that facilitates browsing for users who do not have a fully specified query, but rather, prefer to browse groups of documents whose content the user needs only loosely define to the system.

The Subject Field Codes (SFCs) are based on a culturally validated semantic coding scheme developed for use in (LDOCE), a general purpose dictionary. Operationally, our system tags each word in a document with the appropriate SFC from the dictionary. The within-document SFC frequencies are normalized and each document is represented as a frequency-weighted, fixed-length vector of the SFCs occurring in that document. For retrieval, queries are likewise represented as SFC vectors. The system matches a query SFC vector to the SFC vector of each document in the database. The documents are then ranked on the basis of their vectors' similarity to the query and those documents whose SFC vectors exceed a predetermined criterion of similarity to the query SFC vector can either be displayed to the user immediately or passed on to other system modules for further enrichment.

The SFC vectors represent texts at a more abstract, conceptual level than the individual words in the natural language texts themselves. This addresses the dual problems of synonymy and polysemy. On the one hand, the use of SFCs takes care of the "synonymous phrasing" problem by representing text at a level above the word-level by the assignment of one SFC from amongst 124 possible codes to each word in the document. This means that if four synonymous terms were used within a text, our system would assign each of them the same SFC since they share a common domain which would be reflected by their sharing a common SFC. For example, several documents that discuss the effects of recent political movements on legislation regarding civil rights would have similar SFC vector representations even though the vocabulary choices of the individual authors might be quite varied. Even more importantly, if a user who is seeking documents on this same topic expresses her information need in terms which do not match the vocabulary of any of the documents, her query will still show high similarity to these documents' representations because both the query's representation and the documents' representations are at the more abstract, semantic-field level and the distribution of SFCodes on the vectors of the query and the relevant documents would be proportionately similar across the relevant SFCs.

The other problem with natural language as a representation alternative that has plagued its use in information retrieval is polysemy, the ability of a single word to have multiple senses or meanings. Our Subject Field Coder uses psycholinguistically-justified sense disambiguation procedures [12] to select a single sense for each word. The machine-readable tape of the 1987 edition of LDOCE contains 35,899 headwords and 53,838 senses, for an average of 1.499 senses per headword. The problem is even most serious in regard to the most frequently used lexical items. According to Gentner [13] the twenty most frequent nouns in English have an average of 7.3 senses each, while the twenty most frequent verbs have an average of 12.4 senses each. Since a particular word may function as more than one part of speech and each word may also have more than one sense, each of these entries and/or senses may be assigned different SFCs. This is a slight variant of the standard disambiguation problem, which has shown itself to be nearly intractable for most NLP applications, but which is successfully handled in DR-LINK and allows the system to produce quite reasonable semantic SFC vectors.

We based our computational approach to successful disambiguation on current psycholinguistic research literature which we interpret as suggesting that there are three potential sources of influence on the human disambiguation process: 1) local context, 2) domain knowledge, and 3) frequency data. We have computationally approximated these three knowledge sources in our disambiguator. We consider the 'uniquely assigned' and 'high-frequency' SFCs of words within a single sentence as providing the local context which suggests the correct SFC for an ambiguous word. The SFC correlation matrix which was generated by processing a corpus of 977 Wall Street Journal (WSJ) articles containing 442,059 words, equates to the domain knowledge (WSJ topics) that is called upon for disambiguation if the local context does not resolve the ambiguity. And finally, ordering of SFCs in LDOCE replicates the frequency-of-use criterion. We implement the computational disambiguation process by moving in stages from the more local level to the most global type of disambiguation, using these sources of information to guide the disambiguation process. The work is unique in that it successfully combines large-scale statistical evidence with the more commonly espoused local heuristics. We tested our SFC disambiguation procedures on a sample of twelve randomly selected WSJ articles containing 1638 words which had SFCs in LDOCE. The system implementation of the disambiguation procedures was run and a single SFC was selected for each word. These SFCs were compared to the sense-selections made by an independent judge. The disambiguation implementation selected the correct SFC 89% of the time. This means that a word such as 'drugs', which might refer to either medically prescribed remedies or illegal intoxicants that are traded on the street would be represented differently based on the context of the sentence in which it occurred.

The assignment of SFCs is fully automatic and does not require any human intervention. In addition, this level of semantic representation of texts is very efficient, processing a megabyte of very noisy data in 20 minutes on a Sun4 at normal load. The SFC representation has been empirically tested as a reasonable approach for ranking documents from a very large incoming flux of documents. For the 18th month TIPSTER evaluation, the use of this representation allowed the system to quickly rank 1.14 gigabytes of text in the routing situation that was tested so that all the later-determined relevant documents were within the top 37% of the ranked documents produced by the SFC Module.

However, since the cut-off criterion algorithm which will determine for each individual query how many of the top-ranked documents should be further processed by the remaining modules had not yet been developed, the full system results reported earlier were simply tested against the top 2,000 out of the173,000. Once the algorithm is in place, we will have a reasonable mathematical means for passing on to the RCD modules a ranking which contains all the relevant documents.

Additionally, the SFC vector representation scheme has been experimented with for use on retrospective or ad hoc queries. The SFC vectors are clustered using Ward's agglomerative clustering algorithm [14] to form classes in the document database. The ad hoc queries are likewise represented as SFC vectors and matched to the prototype SFC vector of each cluster in the database. Clusters whose prototype SFC vectors exhibit a predetermined criterion of similarity to the query SFC vector are passed on to other system components for further refinement in representation and matching or can be immediately browsed by the user [15].

A qualitative analysis of the clusters revealed that the use of SFCs combined with Ward's clustering algorithm resulted in meaningful groupings of documents that were similar across concepts not directly encoded in SFCs. Two examples: all of the documents about AIDS clustered together. Secondly, all of the documents about the hostages in Iran cluster ed together even though proper nouns are not included in LDOCE and the word 'hostage' is tagged with the same SFC as hundreds of other terms. What appears to happen with the SFC representation of documents is that relatively equal distributions of words from the same sets of SFCs are found in documents about the same or very similar topics.

8. Proper Noun Interpreter

The Proper Noun (PN) Interpreter [16] was originally developed as a second-level process within the Subject Field Coder Module (but now is an independent module) because our results from earlier testing demonstrated that proper nouns in queries frequently serve as the most important key terms for identifying relevant documents in a database. In addition, some common nouns (e.g. 'developing countries') or group proper nouns (e.g. 'U.S. government') may need to be expanded to their constituent set of proper nouns in order to serve as useful retrieval terms. Therefore, we have implemented several approaches for categorizing and matching proper nouns or their group name in queries to proper nouns in documents. One approach is to expand a group proper noun in a query such as 'U.S. government' to all possible names and variants of entities that comprise the group, or a group common noun such as 'developing countries' in the same manner. Another approach assigns categories from a proper noun classification scheme to every proper noun in both documents and queries to permit proper noun matching at the category level as well as the string matching level.

The Proper Noun Interpreter uses a variety of knowledge bases and processing heuristics to assign a PN category code (e.g. company, person, country) to every proper noun and to index each proper noun within the text so that multiple mentions of the same proper noun entity are all resolved to the same index. Using either the proper nouns themselves or their category codes within the PN Field, allows a range of proper noun matchings to be done. For processing the queries for their proper noun requirements, we have developed a Boolean criteria script which determines which proper nouns or combinations of proper nouns are needed by each query. This requirement is then run against the PN Field of each document to rank documents according to the extent to which they match this requirement. In the recent testing of our system, these values were used to rerank the ranked list of documents received from the SFCoder. The results of this reranking placed all the relevant documents within the top 28% of the database. It should also be noted that the precision figures on the output of the SFC Module + the PN Module produced very reasonable precision results (.22 for the 11-point precision average), even though the combination of these first two modules was not intended to function as a stand-alone retrieval system.

In addition, the Proper Noun Field is a source of extensive information for the later module, the Relation-Concept-Detector, in that many relations useful for producing concept-relation-concept triples have already been determined by the Proper Noun Interpreter and stored in the PN Field.

9. Text Structurer

The purpose of the Text Structuring module in DR-LINK is to delineate the discourse-level organization of each document's contents so that those documents in which the information required by the query is located within the correct discourse component as suggested by the query, can be selected for higher weighting. For example, in newspaper texts, opinions will be found in EVALUATION components, basic facts of the news story will be found in MAIN EVENT components, and predictions will be found in EXPECTATION components. The Text Structurer produces an enriched representation of each document by decomposing it into these smaller, conceptually labelled components. In parallel, the Topic Statement Processor evaluates each topic statement to determine if there is an indication that a particular component in the documents should be more highly weighted when matched to the query representation. For example, indicator-terms in the query such as *predict* or *anticipate* or *proposed* reveal that the time frame of the event being searched for must be in the future, in order for the document to be relevant. Therefore, documents in which this event is reported in a piece of text which has been marked by the Text Structurer as being either EXPECTATION or MAIN, FUTURE would be ranked more highly than those in which this event is reported in a component indicating the event had occurred in the past.

The Text Structurer is based on the News-Text Model for newspaper text, an extended version of the original newspaper text model proposed by van Dijk [17]. The components in the News-Text Model are: CIRCUMSTANCE, CONSEQUENCE, CREDENTIALS, DEFINITION, ERROR, EVALUATION, EXPECTATION, HISTORY, LEAD, MAIN EVENT, NO COMMENT, PREVIOUS EVENT, REFERENCES, and VERBAL REACTION.

Data from a sample set of Wall Street Journal articles were analyzed statistically and translated into computationally recognizable text characteristics to be used by the Text Structurer to assign a component label to each sentence. The main source on which the Text Structurer relies are lexical clues - a set of one, two and three word phrases for each component, chosen because of their frequent occurrences, statistically skewed observed frequency of occurrence in a particular component, and semantic indication of the role or purpose of each component. Operationally, DR-LINK evaluates each sentence in the input text, comparing it to the known characteristics of the prototypical sentence of each component of the News-Text Model, and then assigns a component label to the sentence. The ability of the Text Structurer to correctly decompose and tag news-texts has been shown in preliminary evaluations to be at about the 70% - 80% level [18]. However, further revision and refinement is currently underway with clear indications that the performance can be significantly improved.

The Text Structured document representations will be evaluated by the system to produce more accurate document matches to a query. This need for a specific type of information is recognized during query processing and the query processor maps this need (e.g. CONSEQUENCE components if the query asks for the impact of something; EXPECTATION or FUTURE components if the need is for a predicted or possible event) to its requirement for document matching. The Text Structure Matcher then weights more highly those documents in which the requested information occurs in sentences which have been tagged with the required News-Text component label.

When the system is fully implemented, parallel discourse processing of queries and documents will allow: 1) queries to be evaluated to determine which components of the Text Structure Model will be required in relevant documents, and; 2) an enriched representation of each document to be produced in which each sentence is marked for its appropriate discourse component label. However, our processing is not yet sophisticated enough that we can require that a particular proposition mentioned in the query occur within a particular required component. For example, the matching algorithm at the Text-Structure matching level can require that a document contain a FUTURE component, but we cannot yet require that the acquisition of an American company by a Japanese business occur in that FUTURE component. Also, the discourse processing was done within one module and never used at the final matching and ranking stage. The only empirical testing of the contribution that this module makes to the system's performance was inappropriately done at the18th month TIPSTER testing of the system [19]. For this, the requirement for particular components' presence in the output of the text-structurer was done at the global level as described above and contributed only minimally, as might be expected, to the system's performance.

10. Conclusion

We have developed an information retrieval system that employes various linguistic techniques to process texts semantically and represent them at the conceptual level. Given the task characteristics requiring the handling of semantic constraints in user information needs, we believe that sophisticated processing and rich representation of texts are essential as evidenced by the results of testing the current implementation of the system in the TIPSTER program. Our effort to improve the system is well under way. When we have more complete knowledge bases, the modules are completed, and the various components are more tightly integrated, the performance is expected to improve significantly.

Acknowledgment

The work was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract 91-F136100-000. We thank Ken McVearry and Bob Delzoppo, Coherent Research, Inc. and Chris Khoo, Tracy Lemon, Ming Li, Mary McKenna, Woojin Paik, Ching-cheng Shih, and Edmond Yu at Syracuse University for their contribution, during various time periods, to knowledge base construction and system design and development.

References

[1] Sowa, J. (1984). Conceptual Structures: Information Processing in Mind and Machine. Reading, MA: Addison-Wesley.

[2] Myaeng, S. H. (1992) Using conceptual graphs for information retrieval: a framework for representation and flexible inferencing, Proceedings of Symposium on Document Analysis and Information Retrieval, Las Vegas, March 16-18.

[3] Myaeng, S. H. & Khoo, C. (1992). On uncertainty handling in plausible reasoning with conceptual graphs, Proc. of 7th Workshop on Conceptual Graphs, Las Cruces, NM, July, 1992.

[4] Cook, W. (1989). Case Grammar Theory. Georgetown University Press.

[5] Somers, H. L. (1987) Valency and Case in Computational Linguistics, Edinburgh University Press, Edinburgh.

[6] Myaeng, S. H. & Lopez-lopes, Aurelio (1992). A conceptual graph matching: a flexible algorithm and experiments. Journal of Experimental and Theoretical Artificial Intelligence, Vol. 4, 107-126.

[7] Shafer, G. (1976). A mathematical theory of evidence. Princeton, NJ, Princeton University Press.

[8] Liddy, E.D. & Paik, W. (1991). An intelligent semantic relation assigner. Proceed-

ings of Workshop on Natural Language Learning. Sponsored by IJCAI '91, Sydney, Australia.

[9] Meteer, M., Schwartz, R. & Weischedel, R. (1991). POST: Using probabilities in language processing. Proceedings of the Twelfth International Conference on Artificial Intelligence. Sydney, Australia.

[10] Myaeng, S. H., Khoo, C., & Li, M. (1993) Linguistic Processing of Text for a Large-Scale Conceptual Information Retrieval System, unpublished manuscript.

[11] TREC Conference Workbook, Rockville, MD, Nov, 1992

[12] Liddy, E.D. & Paik, W. (1992). Statistically-guided word sense disambiguation. In Proceedings of AAAI Fall Symposium Series: Probabilistic approaches to natural language. Menlo Park, CA: AAAI.

[13] Gentner, D. (1981). Some interesting differences between verbs and nouns. Cognition and brain theory. 4(2), 161-178.

[14] Ward, J. (1963). Hierarchical grouping to optimize an objection function. Journal of the American Statistical Association, 58, pp. 237-254.

[15] Liddy, E. D., Paik, W. & Woelfuel, J. K. (1992). Use of subject fileld codes from a machine-readable dictionary for automatic classification of documents. Advances in Classification Research: Proceedings of the 3rd ASIS SIG/CR Classification Research Workshop. Medford, NJ: Learned Information, Inc.

[16] Paik, W., Liddy, E.D., Yu, E.S. & McKenna, M. (In press). Interpreting proper nouns for information retrieval. Proceedings of the Human Language Technology Workshop. Princeton, NJ: March, 1993.

[17] van Dijk, T (1988). News analysis: Case studies of international and national news in the press. Hillsdale, NJ: Lawrence Earlbaum Associates.

[18] Liddy, E.D., McVearry, K.A., Paik, W., Yu, E.S. & McKenna, M; (In press). Development, implementation & testing of a discourse model for newspaper texts. Proceedings of the Human Language Technology Workshop. Princeton, NJ: March, 1993.

[19] Liddy, E. D. & Myaeng, S. H. (1993). DR-LINK: 18th Month Progress Report, in Workbook for TIPSTER 18th Month Meeting, Williamsburg, VA, Feb., 1993.