1994

DR-LINK: A System Update for TREC-2

Elizabeth D. Liddy
Syracuse University, School of Information Studies

Sung H. Myaeng
Syracuse University, School of Information Studies

Recommended Citation

Liddy, E.D. & Myaeng, S.H. (1994). DR-LINK: A system update for TREC-2. Proceedings of Second Text Retrieval Conference (TREC-2). National Institute of Standards and Technology.

DR-LINK: A System Update for TREC-2

Elizabeth D. Liddy
Sung H. Myaeng
School of Information Studies
Syracuse University
Syracuse, New York 132100-4100

liddy@mailbox.syr.edu; shmyaeng@mailbox.syr.edu

1. Overview of DR-LINK's Approach

The theoretical goal underlying the DR-LINK System is to represent and match documents and queries at the various linguistic levels at which human language conveys meaning. Accordingly, we have developed a modular system which processes and represents text at the lexical, syntactic, semantic, and discourse levels of language. In concert, these levels of processing permit DR-LINK to achieve a level of intelligent retrieval beyond more traditional approaches. In addition, the rich annotations to text produced by DR-LINK are replete with much of the semantics necessary for document extraction.

The system was planned and developed in a modular fashion and functional modularity has been achieved, while a full integration of these multiple levels of linguistic processing is within reach. As currently configured, DR-LINK performs a staged processing of documents, with each module adding a meaningful annotation to the text. For matching, a Topic Statement undergoes analogous processing to determine its relevancy requirements for documents at each stage. Among the many benefits of staged processing are: improvements and changes can be easily made within any module; the contribution of the various stages can be empirically tested by simply turning them on or off; modules can be re-ordered (as was done within the last six months) in order to utilize document annotations in various ways, and; individual modules can be incorporated in other evolving systems.

The purpose of each of the processing modules will be briefly introduced here (also see Figure 1) in the order in which the system is currently run, with fuller explanations provided in the section below: 1) the Text Structurer labels clauses or sentences with a text-component tag which provides a means for responding to the discourse level Topic Statement requirements of time, source, intentionality, and state of completion; 2) the Subject Field Coder provides a subject-based, summary-level vector representation of the content of each text; 3) the Proper Noun Interpreter and 4) the Complex Nominal Phraser provide precise levels of content representation in the form of concepts and relations, as well as controlled expansion of group nouns and content-bearing nominal phrases; 5) the Relation-Concept Detector produces concept-relation-concept triples with a range of semantic relations expressed via various syntactic classes, e.g. verbs, nominalized verbs, complex nominals, and proper nouns; 6) the Conceptual Graph Generator combines the triples to form a CG and adds Roget International Thesaurus (RIT) codes to concept nodes, and; 7) the Conceptual Graph Matcher determines the degree of overlap between a query graph and graphs of those documents which surpass a statistically predetermined criterion of likelihood of relevance based on ranking by the integrated processing of the first four system modules.

2. Detailed System Description

In the following system description, emphasis is placed on work accomplished within the last year, plus a basic overview description of each module. The more rudimentary processing details of each module plus fuller description of earlier development are available in the TREC-1 Proceedings (Harman, 1993).

2. A. Text Structurer

Since human interpretation of text is influenced by expectations regarding the text to be read, discourse level analysis is required for a system to approximate the same level of meaningful representation and matching. DR-LINK's Text Structurer is based on discourse linguistic theory which suggests that texts of a particular type have a predictable

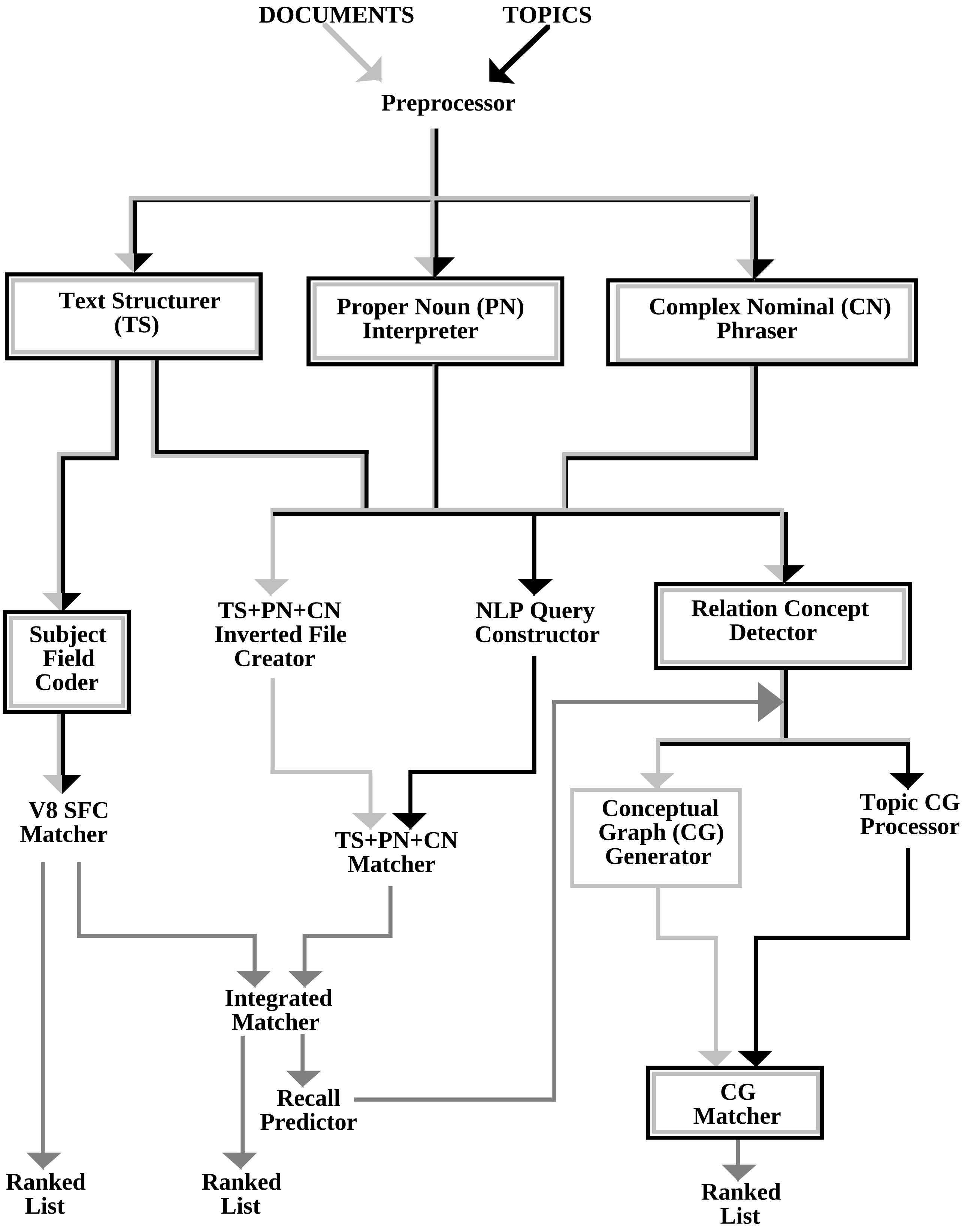

Fig. 1: DR-LINK System

text-level structure which is used by both producers and readers of that text-type as an indication of how and where certain information endemic to that text-type will be conveyed. We have implemented a Text Structurer for the newspaper text-type, which produces an annotated version of a news article in which each clause or sentence is tagged for the specific slot it instantiates in the news-text model, an extension of van Dijk's earlier model (1988). The structural annotations are used to respond more precisely to information needs expressed in Topic Statements, where some aspects of relevancy can only be met by understanding a Topic Statement's discourse requirements. For example, Topic Statement 75, states that:

> *Document will identify an instance in which automation has clearly paid off, or conversely, has failed.*

which contains the implicit discourse requirement that relevant instances should occur in the CONSEQUENCE component of a news article. DR-LINK extracts this requirement from the Topic Statement and will only assign a similarity value for the discourse-level of relevance to those documents in which the sought information occurs in a CONSEQUENCE component.

The current news-text model consists of thirty-eight recognizable components of information observed in a large sample of training texts (e.g. MAIN EVENT, VERBAL REACTION, EVALUATION, FUTURE CONSEQUENCE, PREVIOUS EVENT). The Text Structurer assigns these component labels to document clauses or sentences on the basis of lexical clues learned from text, which now comprise a special lexicon. We considered expanding the lexicon via available lexical resources such as Roget's International Thesaurus or WordNet, but our analysis of these resources suggested that they do not capture the particularities of lexical usage in the sublanguage of newspaper reporting.

The Text Structurer has recently been improved to assign structural tags at the clause level, a refinement which has corrected most of the anomalies that were observed in earlier testings of the Text Structurer. For example, given the new clause-level structuring, the following sentence is correctly interpreted as containing both future-oriented information in the LEAD-FUTURE segment and some nested information regarding a past situation in the LEAD-HISTORY segment.

> <LEAD-FUT> *South Korea's trade surplus*, <LEAD-HIST> *which more than doubled in 1987 to $6.55 billion*, </LEAD-HIST> *is expected to narrow this year to above $4 billion*. </LEAD-FUT>

We have recently implemented new matching techniques which more fully realize the Text Structurer's potential contribution to the system's performance. This was achieved as one outcome of a study which greatly increased our understanding of how text structure requirements in Topic Statements should be used for matching documents to Topic Statements. Analysis of relevant and non-relevant documents retrieved for a test sample of Topic Statements indicated that most of the errors in the Text Structurer's matching were not serious errors, but only slight mismatches in terms of the conceptual definitions of some of the text model's components. This suggested that our model was overly specific for the task of responding to discourse aspects of information requirements, and that matching Text Structure needs from a Topic Statement to structured documents called for a more generalized model. That is, Topic Statement text-structure requirements are not expressed at the same level of specificity at which Text Structure components are recognizable in documents.

Given this, we reduced the matching complexity via a function that maps the thirty-eight news-text components to seven meta-components. These are: LEAD-MAIN, HISTORY, FUTURE, CONSEQUENCE, EVALUATION, ONGOING, and OTHERS. The new approach allows the system to continue to impose the finer-level, 38-component structure on the newspaper articles themselves with excellent precision, but maps this fuller set of text components to the seven meta-components at the matching stage, as the Topic Statements' text structure requirements are coded at the meta-component level. Unofficial experimental results indicate that this new scheme has significantly increased the Text Structurer's contribution to an improved level of precision in the retrieval of relevant documents.

While the Text Structurer module processes documents as described above, the analysis of Topic Statements for their Text Structure requirements is done by the Natural Language Query Constructor (QC) which also analyzes the proper noun and complex nominal requirements of Topic Statements. The QC, as well as the matching and ranking of documents using these sources of linguistic information, is described below.

2. B. Subject Field Coder

The Subject Field Coder (SFC), as reported at TREC-1, has been producing consistently reliable semantic vectors to represent both documents and Topic Statements using the semantic codes assigned to word senses in a machine-readable dictionary. Details regarding this process are reported in detail in Liddy et al, 1993. Our more recent efforts on this module have focused on multiple ways to exploit SFC-based similarity values between document and query vectors. One implementation is the use of the ranked vector-similarity values for predicting a cut-off criterion of potentially relevant documents when the module is used as an initial filter. This is to replace the earlier practice used in the eighteenth month TIPSTER testing, where documents were ranked by their SFC-vector similarity to a query SFC-vector and the top two thousand documents were passed to the CG Matcher, since CG matching is too computationally expensive to handle all documents in the collection. To report the SFC's performance, at that time we reported how far down the ranked list of documents the system would need to process documents in order to get all the judged relevant documents. Although the results were highly promising (all relevant documents were, on average, in the top 37% of the ranked list based on SFC similarity values), this figure varies considerably for individual Topic Statements. Therefore, we needed to devise a method for predicting a priori for individual Topic Statements, the cut-off criterion for any desired level of recall. We first developed a method that could successfully predict a cut-off criterion based on just SFC similarity values. We then extended the algorithm to incorporate the similarity values produced when proper noun, complex nominal, and text structure requirements are considered as well, to produce an integrated ranking based on these varied sources of linguistic information.

The SFC-based cut-off criterion uses a multiple regression formula which was developed on the odd-numbered Topic Statements from 1 to 50 and a training corpus of Wall Street Journal articles. The regression formula takes into account the distribution of similarity values for documents in response to a particular query by incorporating the mean and standard deviation of the similarity value distribution, the similarity of the top-ranked document, and the desired recall level. The cut-off criterion was tested on the held-out, twenty-five Topic Statements. The averaged results, when a user is striving for 100% recall, showed that only 39.65 % of the 173,255 documents would need to be processed further. And this document set, in fact, contained 92% of the judged-relevant documents.

The advantage of the cut-off criterion is it's sensitivity to the varied distributions of SFC similarity values for individual Topic Statements, which appears to reflect how "appropriate" a Topic Statement is for a particular database. For many queries, a relatively small portion of the database, when ranked by similarity to the Topic Statement, will need to be further processed. For example, for Topic Statement forty-two, when the goal is 100% recall, the regression formula predicts a cut-off criterion similarity value which requires that only 13% of the ranked output be further processed, and the available relevance judgments show that this pool of documents contains 99% of the documents judged relevant for that query.

2. C. V-8 Matching

Given the complete modularity of the first four modules in the system, for the twenty-four month TIPSTER testing, we reordered two modules so that Text Structuring is done prior to Subject Field Coding. This allowed us to implement and test a new version of matching which combines in a unique way the Text Structurer and the Subject Field Coder. We refer to this version as the V-8 model, since eight SFC vectors are produced for each document, one for each of the seven meta-categories, plus one for all of the categories combined. The V-8 model, therefore, provides multiple SFC vectors for each document, thereby representing the distribution of SFCs over the various meta-text components that occur in a news-text document. This means, in the V-8 matching, that if certain content areas of the Topic Statement are required to occur in a document in one meta-text component, e.g. CONSEQUENCE, and other content is required to occur in another meta-text component, e.g. FUTURE, this proportional division can be matched against the V-8 vectors produced for each document at a fairly abstract, subject level. For the TIPSTER twenty-four month evaluation, we have experimented with several formulas for combining the similarity values of the multiple SFC vectors produced for each document, including both a Dempster-Shafer combination and a straight averaging. Although official results are not yet available, our internal test results indicate that the combination of Text Structuring and Subject Field Coding produces an improved ranking of documents, especially when using the Dempster-Shafer method.

2. D. Proper Noun Interpreter

Our earlier work with the SFCoder, suggested that the most important factor in improving the performance of this upstream ranking module, would be to integrate the general subject-level representation provided by SFCodes with a level of text representation that enabled more refined discrimination. Analysis of earlier test results suggested that proper noun (PN) matching that incorporated both particular proper nouns (e.g. Argentina, FAA) as well as 'category' level proper nouns (e.g. third-world country, government agency) would improve precision performance. The Proper Noun Interpreter (Paik et al, 1993) that we developed provides: a canonical representation of each proper noun; a classification of each proper noun into one of thirty-seven categories, and; a means for expanding group nouns into their constituent members (e.g. all the countries comprising the Third World). Recent work on our proper noun algorithms, context-based rules, and knowledge bases, has improved the module's ability to recognize and categorize proper nouns to 93% correct categorization using 37 categories as tested on a sample set of 545 proper nouns from newspaper text. The improved performance has a double impact on the system's retrieval performance, as proper nouns contribute both to the downstream relation-concept representation used in CG matching as well as to the upstream proper noun, complex nominal, and text structure ranking of documents in relation to individual queries. Details of processing Topic Statements for their PN requirements and the use of this similarity value in document ranking is described in the later section on the Query Constructor.

2. E. Complex Nominal Phraser

A new level of natural language processing has been incorporated in the DR-LINK System with the implementation of the Complex Nominal (CN) Phraser. The motivation behind this addition was our recognition that either, in addition to proper nouns, or in the absence of proper nouns, most of the substantive content requirements of Topic Statements are expressed in complex nominals (i. e. noun + noun, e.g. "debt reduction", "government assistance", "health hazards"). Complex nominals provide a linguistic means for precise conceptual matching, as do proper nouns. However, the conceptual content of complex nominals can be expressed in synonymous phrases, in a different way than can the conceptual content of proper nouns, which are more particularized. Therefore, for complex nominals, a controlled expansion step was incorporated in the CN matching process in order to accomplish the desired goals of improved recall, as well as improved precision.

For input to the CN Phraser, the complex nominals in Topic Statements are recognizable as adjacent noun pairs or non-predicating adjective + noun pairs in the output of the part-of-speech tagger. Having recognized all CNs, the substitutable phrases for each complex nominal are found by computationally determining the overlap of synonymous terms suggested by RIT and statistical corpus analysis. These processes serve to identify all second order associations between each complex nominal constituent and terms in the database. Second order associations exist between terms that are used interchangeably in certain contexts. The premise here is that if, for example, terms *a* and *b* are both frequently premodified by the same set of terms in a corpus, it is highly likely that terms *a* and *b* are substitutable for each other within these phrases. The use of both corpus and RIT information appears to limit the over-generation that frequently results from automatic term expansion. Ongoing experiments on this new addition to the system will help us further refine the process and will be reported more extensively in the near future.

The terms that exhibit second order associations are compiled into equivalence classes. These equivalence classes provide substitutable synonymous phrases for Topic Statement complex nominals and are used by the matching algorithms in the same manner that the original complex nominals are used. The complex nominals and their substitutes are first used in the upstream matching of Topic Statements to documents as one contributing factor to the integrated similarity value, to be further explained in the section on the Query Constructor.

In addition, each complex nominal and its assigned relation provides a CRC to the RCD module for use in the final round of matching. For that module, semantic relations between the constituent nouns of each complex nominal are assigned manually, using an ontology of forty-three relations. Some example complex nominals plus relation are:

[press] <- (SOURCE) <- [commentaries]
[growth] -> (MEASURE) -> [rate]
[electronic] <- (MEANS) <- [theft]
[campaign] <- (USED_FOR) <- [finances]

The development of the complex nominal CRC knowledge base was an intellectual effort for the twenty four month testing, but our current task is the full automation of the semantic relation assignment. Although difficult, our experience with the intellectual process has encouraged us to pursue appropriate NLP-based machine-learning techniques which will enable the system to automatically recognize and code semantic relations in complex nominals.

In CG matching, the existence of both case-frame relations and complex nominal relations make it possible for the system to detect conceptual similarity even if expressed in different grammatical structures, such as a verb + arguments in a Topic Statement and a complex nominal in a document, e.g.:

"reduce the debt" = [reduc*] -> (OBJECT) -> [debt]
"debt reduction" = [debt] <- (OBJECT) <- [reduc*]

To achieve the fullest exploitation of relational information despite grammatical realization, a further step was necessary in order to match on CRCs produced by verb-based analysis and CRCs produced by complex nominal analysis. This required the determination of the degree of relation-similarity across the two relation sets. There are approximately sixty relations used in case frames, while there are approximately forty relations used in complex nominals. A relation-similarity table was constructed that assigns a degree of similarity between twenty-eight pairs across the two grammatically-distinguished sets, and a degree of similarity between pairs within the same set. The relation-similarity table is used in the final CG matching to allow concepts that are linked by a relation in a document that is different from the relation that links the same two concepts in the Topic Statement, to still be awarded some degree of similarity. The quality and appropriateness of the similarity table will be determined by the results of the twenty-four month testing which will also provide empirical evidence of the Complex Nominal Phraser's impact on performance. Sample runs have indicated that the inclusion of complex nominals has a strongly positive impact on our results in both of its incorporations in the system.

2. F. Natural Language Query Constructor

We have implemented a Natural Language Query Constructor (QC) for DR-LINK which takes as input a Topic Statement which has been pre-processed by straight-forward techniques, such as part-of-speech tagging as well as SGML-tagging of the meta-language which reflects the typical request-presentation language used in Topic Statements (e.g. "A relevant document will ..." or "To be relevant..."). The QC produces a query which reflects the appropriate logical combinations of the text structure, proper noun, and complex nominal requirements of a Topic Statement. The basis of the QC is a sublanguage grammar which is a generalization over the regularities exhibited in the Topic, Description, and Narrative fields of the one hundred fifty TIPSTER Topic Statements. It should be noted that the sublanguage grammar, with minor modifications, is capable of handling non-TIPSTER queries, so its generalized utility is promising. Earlier work (Liddy et al, 1991) demonstrated that the sublanguage approach is an effective and efficient approach to natural language processing tasks within a particular text-type, here Topic Statements.

For the twenty-four month runs, the QC sublanguage grammar detects the required logical combination of text structure components, proper nouns, and complex nominals. These are the specific entities which we consider to be particularly revealing indicators of relevant documents. In most cases, matching on these classes produces high-precision ranked results, although there are some instances in which single common nouns may also be needed. After analyzing the twenty-four month results, we will determine whether to expand the range of linguistic types which can be used to instantiate the variables in the QC's logical assertions.

The QC sublanguage grammar relies on function words (e.g. conjunctions, prepositions, relative pronouns), meta-level phrases (e.g. "such as", "examples of", "as well as"), and punctuation (e.g. commas, semi-colons) to recognize and extract the relevancy requirements of Topic Statements. These linguistic features serve as clues to the 'organizing' structure of a Topic Statement and present each Topic Statement's unique thematic content in a recognizable frame. The QC sublanguage interprets a Topic Statement into pattern-action rules which are used to reduce each sentence in a Topic Statement into a first order logic assertion, reflecting the boolean-like requirements of Topic Statements, including NOT'd assertions. In addition, definite noun phrase anaphors are recognized and resolved by sublanguage grammar processing rules.

2. G. Integrated Matcher

Each logical assertion produced by the QC for a Topic Statement is evaluated against the entries in the document inverted file and a weight is assigned to each segment of text (either a clause or a sentence) which has any similarity. The weighting scheme we are currently using evolved from iterative testing. Each segment of text is indexed in the inverted file with a text structure component label and will be assigned a weight if it contains any proper nouns or complex nominals that match the Topic Statement's requirements. The following weights are assigned:

proper noun	=	1.00
complex nominal	=	1.00
proper noun category	=	0.50

This means, for example, that if, in response to the following requirement from a Topic Statement:

> *A relevant document will provide data on Japanese laws, regulations, and/or practices which help the foreigner understand how Japan controls, or does not control, stock-market practices which could be labeled as insider trading.*

a document text-segment contains 'Japanese law', and 'stock-market practice' (or one of its synonymous phrases), and 'insider trading' (or one of its synonymous phrases), that segment is assigned a preliminary value of 3.00. Depending on which field in the Topic Statement the assertion came from, and whether the document text-segment matches the Topic Statement's Text Structure requirement, the preliminary value will be multiplied by one of the following co-efficients:

Topic field and **required** Text Structure component	=	1.00
Desc, Narr, or Concept field and **required** Text Structure component	=	0.75
Topic field and **non-required** Text Structure component	=	0.50
Desc, Narr, or Concept field and **non-required** Text Structure component	=	0.25

So if 'Japanese law' and 'stock-market practice' and 'insider trading' were conceptual requirements from a Topic field assertion that also required them to occur in an EVALUATION or LEAD-MAIN text component, and they occurred in a document text segment which has been tagged by the Text Structurer as EVALUATION, the value of 3 would be multiplied by 1; whereas if that assertion came from the Description field in the Topic Statement and the three required phrases occurred in a document text segment labelled CONSEQUENCE by the Text Structurer, the value of 3 would be multiplied by .25.

Since the QC interprets each sentence in the Topic, Description, Narrative, and Concept fields in a Topic Statement, multiple, sometimes overlapping, sometimes repetitive assertions are produced for a single Topic Statement. In the current implementation, each of these Topic Statement assertions is compared to the inverted document file, and the highest similarity value for a single assertion in the document is used as that document's integrated similarity value for that Topic Statement.

The similarity value which results from the QC module matching is combined with the SFC similarity value of the document, and an integrated similarity score for each document is produced. This similarity value can be used in several ways. Firstly, the two similarity values can be used to provide a full ranking of all the documents which takes into account the lexical, semantic and discourse sources of linguistic information in both documents and queries. Secondly, it can serve as input to a filter which uses a more complex version of the original cut-off criterion to determine how many documents should be further processed by the system's final modules.

For the Integrated Matcher to produce a combined ranking, each document's similarity value for a given Topic Statement can be thought of as being composed of two elements. One element is the SFC similarity value and one element is the similarity value that represents the combined proper noun, complex nominal, and text structure similarities. Additionally, the system will have computed the regression formula, the mean, and standard deviation of the distribution of the SFC similarity values for the individual Topic Statement. Using these statistical values, the system produces the cut-off criterion value. Since we know from the eighteen-month results, that 74% of the relevant documents had what we refer to as a k-value (then PN value; now PN, CN, TS values) and the remaining 26% of the relevant documents had no k-value, we use this information to predict what proportion of the predicted relevant documents should come from which segment of the ranked documents for full recall. The combined ranking can be envisioned as consisting of four segments, as shown in Figure 2.

```
                                  -----------------------------
Docs. having a k-value            |
& an SFC value                    |     Group 1
above the cut-off                 |
                                  ---------  cut-off criterion SFC similarity value
Docs. having a k-value            |
& an SFC value                    |     Group 2
below the cut-off                 |
                                  -----------------------------
Docs. having no k-value           |
& an SFC value                    |     Group 3
above the cut-off                 |
                                  ---------  cut-off criterion SFC similarity value
Docs. having no k-value           |
& an SFC value                    |     Group 4
below the cut-off                 |
                                  ----------------------------
```

Fig. 2: Schematic of Segmented Ranks from SFC & Integrated Ranking (k-value)

Four groups are required to reflect the two-way distinction mentioned above. The first distinction is between those groups which have a k-value and which should contain 74% of the relevant documents and those documents without a k-value, which should contribute 26% of the relevant documents. The second distinction is between those documents whose SFC similarity value is above the predicted cut-off criterion and those whose SFC similarity value is not.

When a cut-off criterion is the application desired, the system will produce the ranked list in response to a desired recall level, by concatenating the documents above the appropriate cut-off for that level of recall from Group 1; then documents above the appropriate cut-off for that level of recall from Group 3. However, since our test results show that there is a potential 8% error in the predicted cut-off criterion for 100% recall, we use extrapolation to add the appropriate proportion of the top ranked documents from Group 2 to Group 1, before concatenating documents from Group 3. These same values are used to produce the best end-to-end ranking of all the documents using the various segments.

Document ranks are produced by the Integrated Matcher and the cut-off criterion is used either by an individual user who requires a certain recall level for a particular information need, or, as in the twenty four month TIPSTER test situation, by the system to determine how many documents from the Integrated Matcher ranking will be passed on to the final modules for further processing.

2. H. Topic Statement Processing for Conceptual Graph Generation

The processing of topic statements for CG generation does not make use of the output of the Natural Language Query Constructor, but instead the current system first applies the same RCD and CG generator modules to produce topic statement (TS) CGs. Several TS-specific processing requirements have been identified, some of which have been implemented as post-processing routines and others are under development.

- Elimination of concept and relation nodes corresponding to contentless meta-phrases (e.g. "Relevant document must identify ..."). If both of the concept nodes in a concept-relation-concept triple belong to a meta-phrase, the CRC is ignored. When only one of them is a meta-phrase concept, the triple is not removed blindly unless the other concept occurs in another triple.

- Handling of negated parts of topic statements. The weights are adjusted in such a way that an occurrence of the negated concept in a document will contribute to the negative evidence that the document will be relevant. In effect, the two weights for the concept are switched.

- Automatic assignment of weights to concept and relation nodes. There are several factors we consider: the conventional way of determining the importance of terms using inverse document frequency (IDF) and total frequency; the location of terms occurring in topic statements; the part of speech information for each term; and indications in the topic statement sublanguage (e.g. the document MUST contain ...). Although we have implemented a program that tags individual words with the degree of importance based on the sublanguage patterns, we assigned concept weights based on IDF values of terms in the collection for the evaluation, due to time constraints.

- Merging common concept appearing in different sections of topic statements. Although it is not safe in general to assume that two concepts sharing the same concept name actually refer to the same concept instantiation and merge them blindly, we have observed that this is not the case in the topic statements. In fact, we believe that it is desirable to merge CG fragments using common concept nodes. This is an important process that eliminates undesirable effects on scoring. Without this, a document containing a concept occurring repeatedly in <desc>, <narr>, and <con> fields would be ranked unnecessarily high (or low if it is negated) because each occurrence of the concept would make an independent contribution to the overall score.

Since an integrated automatic topic processing module was not available, the mechanical aspects of the process were hand-simulated with some parts done automatically and other done manually.

2. I. Relation Concept Detector (RCD)

The output of the Complex Nominal Phraser and the Proper Noun Interpreter modules described above provide concept-relation-concept triples directly to the Relation-Concept Detector (RCD) module. In addition, the following RCD handlers are operative.

One of the more distinct aspects of the DR-LINK system is its capability of extracting and using relations in the final representation of documents and topic statements in their CG representations. This module provides building blocks for the CG representation by generating concept-relation-concept triples based on the domain-independent knowledge bases we have been constructing with machine-readable resources and corpus statistics. In this module, there are several handlers that are activated selectively depending on the input sentence.

2. I. 1. Case Frame (CF) Handler

The main function of the CF Handler is to generate concept-relation-concept triples where one of the concepts comes typically from a verb. It identifies a verb in a sentence and connects it to other constituents surrounding the verb. Since the relations (about 50 we use currently) included in our representation are originated from the theories of linguistic case roles (Somers, 1987, and Cook, 1989) and are all semantic in nature, this module consults the knowledge base containing 13,786 case frames we have constructed, each of which prescribes a pattern involving a verb and the corresponding concept-relation-concept triples.

Given a set of case frames for different senses of `decline', for example,

```
(decline 1      ((PATIENT subject ? obligatory)))

(decline 2      ((AGENT subject human obligatory)
                    (PATIENT object ? optional)))

(decline 3      ((AGENT subject human obligatory)
                    (ACTIVITY infinitive ? obligatory)
                    (link infinitive subject AGENT)))
```

AGENT, PATIENT, and ACTIVITY are the relations that connect the verb to other constituents. The second components (e.g. subject) prescribe the syntactic categories of the constituents and the third components (e.g. human) semantic restrictions that the subject and object should satisfy. The last components (e.g. obligatory) indicate whether the constituent must exist in a sentence in order for the particular case frame to be instantiated. The last line of the third case frame instructs the CF handler to link the subject to the infinitive verb with the AGENT relation. This kind of linking instructions allow the CF handler to produce triples containing non-verbal constituents.

The CF Handler selects the best case frame by attempting to instantiate each case frame and determine which one is satisfied most by the sentence at hand. This can be seen as a sense disambiguation process using both syntactic and semantic information. The semantic restriction information contained in the case frames were obtained from LDOCE, and when the sentence is processed, the CF handler also consults LDOCE to get semantic restriction information for individual constituents surrounding the verb in the sentence and compares it with the restrictions in the case frames of the verb as a way to determine which case frame is likely to be the correct one.

With the following sentence fragment,

> ... the chairman declined to elaborate on the disclosure ...

the CF handler chooses the third case frame and produces

```
[decline] -> (AGENT) -> [chairman]
[decline] -> (ACTIVITY) -> [elaborate]
[elaborate] -> (AGENT) -> [chairman]
```

In the current implementation, the input text to the CF handler is first tagged with part-of-speech information and bracketed for constituent boundaries. BBN's POST tagger (Metter et al., 1991) has been used to attach a part-of-speech tag to individual words. The constituent boundary bracketer we developed then marks boundaries of grammatical constituents such as infinitives, noun phrases, prepositional phrases, clauses, etc.

At the time of writing, the case frame knowledge base contains 13, 786 case frames, of which 13,444 are for all the verb entries (5,206) in LDOCE, and the rest are for 342 verbs that appear in the Wall Street Journal collection but are not in the LDOCE as a headword. While we have constructed case frames for most of the phrasal verbs in LDOCE, the capability of processing phrasal verbs has not been implemented in the current CF Handler.

2. I. 2. Nominalized Verb (NV) Handler

The nominalized verb handler has been implemented for the DR-LINK system we ran for the TIPSTER 24th month evaluation. Its main function is to consult the NV case frames to identify a NV in a sentence and create concept-relation-concept triples based on the rule. At the same time, it converts the NV into its verb form. In this way, we can allow for a match between a CG fragments generated from a phrase containing verb and another fragment generated from a noun phrase containing the corresponding nominalized verb. For example, the NV Handler converts the sentence fragment

> ... the company's investigation of the incident ...

into

> [investigate] -> (AGENT) -> [company]
> [investigate] -> (PATIENT) -> [incident].

This process is much more than a sophisticated way of performing stemming in that we canonicalize concept-relation-concept triples rather than just concept nodes.

For NV processing, 15, 053 case frames have been generated for 1,593 nominalized verbs. Most of the case frames for NVs were automatically generated from the corresponding verb case frames. This process was also facilitated by identifying potential NVs from LDOCE.

No explicit testing of the impact of NVs in information retrieval has been done yet although we have convinced ourselves with anecdotal evidence that this would improve the retrieval performance. More semantic processing of nominalized verbs in determining the relations to the surrounding constituents is on the future research agenda. More rigorous study on the impact of NVs on information retrieval should be done, too.

2. I. 3. Noun Phrase (NP) and Prepositional Phrase (PP) Handler

The noun phrases that are not handled by the complex-nominal handler or by the nominalized verb handler are analyzed so that the head noun is connected to the concepts outside the noun phrase (e.g. a verb concept in the CF Handler). In addition, this module identifies individual concepts corresponding to adjectives and other nouns in a compound noun and connects them with CHARACTERISTIC, ATTRIBUTE, or LINK relations. LINK is the most generic relation in our system.

Once noun phrases are handled this way, this module handles prepositional phrases by connecting the head noun concept of the noun phrase to the preceding constituent (e.g. a verb or a noun). The preposition attachment problem is a difficult one, and the current implementation takes the simple- minded approach with general relations such as LINK, which can match with many of other semantically more specific relations. Our preliminary analysis indicates that this approach correctly handles about 75% of the prepositional phrase cases in the Wall Street Journal collection. More accurate and finer-level processing will be done with more semantically oriented rules that check the semantic restrictions and use more specific relations. The role of this handler will be diminished when we process phrasal verbs as part of the CF handler, for which we have constructed case frames.

2. I. 4. Ad-hoc Handler

This module looks for lexical patterns not covered by any of the other special handlers discussed above. Its processing is also driven by its own knowledge base of patterns to infer relations between concepts. For example, a sentence fragment

> ... bought the item for the purpose of satisfying ...

contains a pattern

> [VERB] ... for the (ADJ) purpose of [NP]

in the knowledge base, and hence results in a triple

[buy] -> (GOAL) -> [satisfy]

The knowledge base contains a small number of simple patterns involving BE verbs and more than 350 pattern rules for phrasal patterns across phrase boundaries, by which important relations are extracted. The pattern rules specify certain lexical patterns and the order of occurrences of words belonging to certain part-of-speech categories, and the concept-relation-concept triples to be generated. These patterns require a processing capability no more powerful than a finite state automaton. Due to the time constraints, however, the current ad-hoc handler has not been generalized to process all the patterns, and about 30% of the patterns in the knowledge base are recognized and handled correctly.

2. J. Conceptual Graph (CG) Generator

After individual RCD modules have generated concept-relation-concept triples for a document, the CG generator merges them to form a set of conceptual graphs, each corresponding to a clause in most cases. Since more than one handler can generate different triples for the same concept pairs (e.g. a prepositional phrase handled by the CF handler and the NP/PP handler) based on independently constructed rules and on independent processes, a form of conflict resolution is necessary. In the current implementation, we simply order the execution of different handlers based on the general quality of the rules and the resulting triples so that more reliable handlers have higher precedence.

The concept nodes in the resulting CGs can not only contain general concept names but also some instantiations (referents) of the concepts. Such a concept can be derived either from a proper noun such as a company name or from a sub-ordinate clause. In the latter case, the instantiation is a CG itself to produce a CG like

[country: {US}] <- (SOURCE-OF-INFO) <- [C#: [[pact] <- (PATIENT) <- ...]

In the current implementation, concepts with the same instantiation are merged across sentences to form a larger CG, but concept with the same label but without any referents across sentences are treated as separate concepts and are not merged. A pronoun resolution method is being implemented to merge a pronoun to its antecedent as a way to increase the connectivity of CGs and hence increase the usefulness of relation nodes.

As a way to make our current representation more "conceptual", we have implemented a module that adds RIT (Roget's International Thesaurus) codes to individual concept nodes so that the label on the nodes is not a word but a position of the hierarchy of RIT. The lowest level position beyond individual lexical items in the RIT hierarchy is called a semi-colon group consisting of several terms within the delimiter of semi-colons, which represents a concept.

The mapping from a word (called target) in text to a position in RIT requires sense disambiguation, and our approach is to use the words surrounding the target word as the context within which the sense of the target word is determined and one or more RIT codes are selected. The algorithm selects minimal number (i.e. one or more) of RIT codes, not just the best one, for target words since we feel that some of the sense distinctions made in RIT are unnecessarily subtle, and it is unlikely that any attempts to make such fine distinctions would be successful and hence contribute to information retrieval.

We have produced RIT-coded documents and topic statements for the San Jose Mercury collection and the routing queries. All the concept nodes derived from nouns now have RIT codes selected using the surrounding text as the context. Those concept nodes derived from verbs also have RIT codes but in a different way. Instead of using the surrounding text as the context and trying to disambiguate senses (we concluded that this method is not reliable for verbs), we first assign RIT codes to each sense of LDOCE verb entries using the same method. In this case the context become the definition text in LDOCE. Once we select the right case frame by Case Frame Handler while text is processed, the RIT codes attached to the case frame are automatically assigned to the target verb.

2. K. Conceptual Graph (CG) Matcher

The main function of the CG matcher is to determine the relevance of each document against a topic statement CG and produce a ranked list of documents as the third and final output of the system. Using the techniques necessary to model plausible inferences with CGs (Myaeng and Khoo, 1992), this module computes the degree to which the topic statement CG is covered by the CGs in the document (see Myaeng and Liddy (1993) and Myaeng & Lopez-lopez (1992) for details).

While the most obvious strength of the CG approach is its ability to enhance precision by exploiting the structure of the CGs and the semantics of relations in document and topic statement CGs, and by attempting to meet the specific semantic constraints of topic statements, we also attempt to increase recall by allowing flexibility in node-level matching. Concept labels can be matched partially (e.g. between `Bill Clinton' and `Clinton'), and both relation and concept labels can be matched inexactly (e.g. between `aid' and `loan' or between `AGENT' and `EXPERIENCER'). For both inexact and partial matches, we determine the degree of matching and apply a multiplication factor less than 1 to the resulting score. For inexact matching cases, we have used a relation similarity table that determines the degree of similarity between pairs of relations. Although this type of matching slows down the matching time, we feel that until we have a more accurate way of determining the conceptual relations and a way to represent at a truly conceptual level (e.g. our attempt to use RIT codes), it is necessary. More importantly, the similarity table reflects our ontology of relations and allows for matching between relations produced by different RCD handlers whose operations in turn are heavily dependent on the domain-independent knowledge bases.

We have done a series of matching experiments internally to evaluate various strategies in CG matching/scoring and document representation with the goal of selecting the best one for the final TIPSTER 24th month runs. The first question we had was how to "normalize" the score assigned to a document based on the current scoring scheme. As described above, the scoring algorithm is query-oriented in the sense that the score reflects to what extent the query CG is covered by the document CG. While this approach is theoretically justifiable, one potential drawback is that a document containing the entire query CG is not ranked higher than one that contains fragments of the query CG scattered in the document as long as they cover the same query CG. That is "connectivity" or `coherence" of matching document CG is not fully taken into account.

With the intuitive notion that the number of matching CG fragments in a document would be inversely proportional to "connectivity", we have been experimenting with various normalization factors that are a function of the number of matching CG fragments. At the time of writing, our experimental data show that when we consider 12 sentential CGs as a unit (called "paragraph") and use the number of units containing one or more matching CG fragments in the normalization function, we obtain the best result. Among all the functions we have tried, the best normalization factor we have found experimentally so far is:

$$1.05 ^ {(1-x)}$$

where x is the number of text units that contain one or more matching CG fragments. When this is combined with the maximum of the scores assigned to individual "paragraph" as follows:

$$S*1.05^{(1-x)} + 0.4*M$$

where S is for the unnormalized score and M for the maximum "paragraph" score, we obtained the best results. Since we determined the constants incrementally, it is entirely possible that different combination of the constants can give better results. It is relatively clear based on these experiments that the first or the second term alone are always inferior to the combination. The number of sentential CGs for "paragraphs", 12, seems also pretty stable.

We have produced TIPSTER runs using the RIT-coded documents and topic statements. The current matching program attempts to match on RIT codes only when the concept names (words) don't match. Because of this conservative approach, the RIT codes do not block a match between two different polysemous words and thus have any direct impact on the word ambiguity problems in IR. With the disambiguation process employed when RIT codes are chosen for a noun or verb, however, the net effect is analogues to term expansion with sense disambiguation. It should be noted that since RIT codes are used for both document and query concepts, this amounts to sense-disambiguated term expansion on both queries and documents.

While the original motivation was to represent documents and topic statements at more conceptual level using RIT codes, we are also testing the effectiveness of RIT-based term expansion in IR environments. Using the scheme we have developed for term clustering using contextual information in the corpus (Myaeng & Li, 1992), we have three methods to evaluate: RIT-based expansion, term-cluster based expansion, and a combination of the two so that we can eliminate the problem of using a general- purpose thesaurus and the errors made by the term-clustering method.

For TIPSTER evaluation, we have submitted two sets of four runs: one with RIT codes and the other without them. Each set consists of three runs for different scoring schemes and the last one for the combination of the three runs which appears to produce the best result in our internal experiment.

3. Test Runs

The DR-LINK group elected to put their efforts into continued work for the twenty-four month TIPSTER testing, and as a result we lost our opportunity to have TREC-compatible results to discuss at this time. Although our twenty-four month TIPSTER runs have been submitted, many of our top-ranked documents were not amongst those submitted by TREC participants, so it is virtually impossible to make even unofficial reports on our system's performance. We trust that in the near future there will be some comparable groups and/or runs to measure ourselves against after the results from both TIPSTER and TREC-2 are available.

4. Summary

As the above descriptions should convey, we have made a great deal of progress in the development and integration of the DR-LINK System since TREC-1. Unfortunately, the absence of quantified results of our performance limits our convincing power. However, we are pleased to have demonstrated that a system implementation of our original notion of integrating multiple levels of linguistic processing so that retrieval can be conducted at a conceptual rather than word-based level is nearly achieved.

Many rich research and implementation ideas remain to be explored in all of the DR-LINK modules, particularly those which have only been in existence for a few months.

5. Acknowledgments

The research reported herein was funded by ARPA's TIPSTER Program. We are grateful to both Longmans' for access to the machine readable tape of LDOCE and to BBN for the loan of their POST tagger.

Without the untiring efforts of the following individuals, neither the DR-LINK System nor our very promising results would have been possible: Margot Clark, Bob Del Zoppo, Saleh Elmohammed, Chris Khoo, Tracey Lemon, Ming Li, Mary McKenna, Ken McVearry, Carin Obad, Woojin Paik, Ching-cheng Shih, Joe Woelfel, Edmund Yu, Ahsan Zhia.

References

Cook, W. (1989). Case Grammar Theory. Washington, D.C. : Georgetown University Press.

Harman, D. (Ed.) (1993). The first Text REtrieval Conference (TREC-1). National Institute of Standards and Technology.

Liddy, E.D., Jorgensen, C.L., Sibert, E. & Yu, E.S. (1991). Sublanguage grammar in natural language processing. Proceedings of RIAO '91 Conference. Barcelona.

Liddy, E.D., Paik, W. & Yu, E.S. (1993). Document filtering using semantic information from a machine readable dictionary. Proceedings of the ACL Workshop on Very Large Corpora.

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

Myaeng, S, H. & Khoo, Chris (1992). On uncertainty handling in plausible reasoning with conceptual graphs. Proceedings of the 7th Conceptual Graphs Workshop. Las Cruces, NM.

Myaeng, S. H. & Li, Ming (1992). Building a database-specific concept hierarchy for information retrieval by acquiring lexical semantics from a corpus. Proceedings of the First International Conference in Information and Knowledge Management. Baltimore, MD.

Myaeng, S. H. & Liddy, E. D. (1993). Information retrieval with semantic representation of texts. Proceedings of Symposium on Document Analysis and Information Retrieval. Las Vegas.

Myaeng, S. H. & Lopez-Lopez, A. (1992). Conceptual graph matching: A flexible algorithm and experiments. Journal of Experimental and Theoretical Artificial Intelligence. Vol. 4, 107-126.

Paik, W., Liddy, E.D., Yu, E.S., & McKenna, M. (1993). Categorizing and standardizing proper nouns for efficient information retrieval. Proceedings of the Workshop on Acquisition of Lexical Knowledge from Text. Association for Computational Linguistics. pp. 154-60.

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

van Dijk, T. A. (1988). News as discourse. HillsdaleLawrence Erlbaum Associates.