Interpreting Nominal Compounds for Information Retrieval*

Linda S. Gay**
Computer and Information Science Department
University of Massachusetts

COINS Technical Report 88-86
October 3, 1988

Abstract

A nominal compound is a group of two or more nouns that together forms a noun, such as: *computer performance evaluation techniques.* Interpreting nominal compounds is important for any natural language program because compounds occur frequently in text. It is also important in information retrieval because user queries contain many nominal compounds and compounds occur in different surface forms in documents.

This report addresses the problem of determining the relationships between the words in a compound. For example, the relationship in *data analysis* is the *object* relationship, while in *silk shirt* it is the *made-of* relationship. Knowledge about the world is essential in determining the relationships. For instance, the knowledge that newspapers are delivered and that a boy is able to deliver things is necessary to interpret *newspaper boy* as: *a boy who delivers newspapers.*

This report also investigates the possibility of using the interpretation of nominal compounds to improve the performance of information retrieval systems. The results indicate that techniques using simple extensions to standard statistical retrieval methods might be more cost-effective in increasing performance than techniques using the interpretation of nominal compounds.

---

*This report is also published as GE CRD Report #88CRD289.

**This work was supported by the General Electric Company while the author was a Summer Intern at the General Electric Research and Development Center in Schenectady, NY.

Contents

INTERPRETING NOMINAL COMPOUNDS FOR INFORMATION RETRIEVAL

Linda S. Gay

1 INTRODUCTION

A nominal compound is a group of two or more nouns that together forms a noun, such as: *control mechanism, information retrieval problems*, and *computer performance evaluation techniques*. Because compounds occur frequently in text, any natural language program must be able to interpret them.

Knowledge about the world is essential in interpreting compounds. For example, the knowledge that newspapers are delivered and that a boy is able to deliver things is necessary to interpret *newspaper boy* as: *a boy who delivers newspapers*.

In interpreting compounds, the last noun in the compound, called the *head* noun, describes the object that the compound expresses. For example, the compound, *information retrieval problem*, describes a particular type of *problem*. The other nouns in the compound modify the head noun in some way. However, the modifiers might be nominal compounds themselves, as in *information retrieval problem*.

This research focuses on finding general mechanisms to determine the relationships among the words in a compound because of the importance of this problem in information retrieval. Designing and coding a knowledge base and interpretation algorithm identified: 1) the features of a knowledge base that are necessary in order to interpret compounds, and 2) a general algorithm that can interpret a majority of compounds.

The interpretation of compounds could potentially improve the precision and recall of information retrieval systems. Understanding the relationship between the words in *data structure* could increase recall by retrieving documents with this compound in different surface forms and documents using synonyms for words in the compound, as in the following phrases:

structures *for* **data**
a variety of **data representations**

Knowing the relationship in *data structure* could also increase precision by filtering out documents containing the words from the compound but not expressing the same concept as in the compound, as in the phrase:

heavily **structured data**

An experiment shows the possibility of increasing precision and recall in these ways.

The rest of this section discusses:

- the problems in interpreting compounds,
- the type of knowledge necessary to interpret compounds,
- the interpretation of compounds,
- the application of this work to information retrieval, and
- the approach taken in this work.

Finally, the last section gives an overview of the rest of this report.

1.1 Problems in Interpreting Nominal Compounds

In interpreting nominal compounds, the following problems arise:

- Determining the relationship(s) between the nouns in a compound,
- Parsing a compound, and
- Disambiguating the word senses of the nouns in a compound.

The first problem is the focus of this report - determining the relationships between the nouns in a compound. In the compound *silk shirt*, the relationship is *made-of*; in *steam locomotive*, the relationship is *powered-by*; and in *milk bottle*, the relationship is *container-for*. The syntax of the compound does not help in determining these conceptual relationships, rather semantic and pragmatic knowledge is necessary. For example, the knowledge needed to interpret *milk bottle* is that a bottle can contain liquids and that milk is a liquid.

Determining the relationships in nominal compounds can be extended to noun modification in general. The knowledge necessary to determine the relationships in *cotton dress* and *red dress* is the same. Both involve information about *dresses* - in the former case, that dresses are made of some kind of material, in the latter, that dresses have color.

Two separate problems exist in parsing compounds. The first problem is identifying a compound while parsing text. For example, the sentence,

The cpu signal interrupts transfer activity[1]

could contain many compounds including: *cpu signal, cpu signal interrupts, cpu signal interrupts transfer activity, signal interrupts, signal interrupts transfer activity, interrupts transfer* and *transfer activity*. Constraints on the syntax of sentences limits the parses of this sentence to two:

- *[The cpu signal] [interrupts] [transfer activity].*
- *[The cpu signal interrupts] [transfer] [activity].*

Thus, the sentence could either contain the two compounds, *cpu signal* and *transfer activity*, or the sentence could contain the one compound, *cpu signal interrupts*. Arens[1] discusses the problem of identifying nominal compounds in sentences in more detail.

The second problem in parsing compounds is to determine how to parse a compound into constituents. This problem only applies to compounds with more than two words. The compound, *cpu signal interrupts*, has two possible parses: *[[cpu signal] interrupts]* and *[cpu [signal interrupts]]*. Section 2.3 discusses this problem in more detail.

Disambiguating word senses involves mapping from the nouns in a compound to the concepts that the noun refers to. Section 3.1.5 briefly discusses this problem.

The three problems in interpreting compounds are not separate problems, rather, each is dependent upon the others. To determine, for example, that the relationship between *can* and *cover* in *plastic cat food can cover*[*] is *a cover for a can*, a system must determine that the word sense for *cover* is *an* **object** *that covers something*, not *the* **act** *of covering*, and that the sense of *can* is a *metal container*, not the *auxiliary verb*. Furthermore, to know that *can cover* is a constituent phrase of the compound, a system must determine that a possible relationship exists between these words. Thus, each of these problems helps to constrain the other problems resulting in a highly interactive process.

1.2 Knowledge Representation

World knowledge is essential in interpreting nominal compounds. To understand a phrase such as *plastic cat food can cover*, world knowledge prevents the consideration of nonsense parses and makes the phrase comprehensible. The interpretation depends on facts like cats eat food, cans have covers, can covers may be made of plastic, etc. A mechanism for interpreting compounds must have access to this kind of knowledge.

The knowledge base must contain concepts and relationships between those concepts to facilitate the interpretation of compounds. The concepts correspond to events and entities in the world, such as *cat, plastic, cat*, etc. They are related to one another through typical *case-frame roles* such as *actor* and *instrument*. The concepts themselves determine which roles are associated with which concepts. For example, the *cat* event has the following roles: 1) the *actor* role (the person who is eating), and 2) the *object* role (what is being eaten). Other roles associated with this event are the *location* and *time* of the event, and the *instrument* used in

[*] I first heard this compound from Wendy Lehnert during the Natural Language Processing course at the University of Massachusetts.

the event. Roles are able to capture the underlying relationship between nouns in a compound. Thus, in the compound, *data analysis*, the relationship is an *object* relationship: the *analysis* of *data*.

Determining the role that exists between words in a compound aids in recognizing the compound in different surface forms. This recognition is important in information retrieval. For example, knowing that the role in *performance evaluation* is the *object* role enables one to recognize the same concept in the forms: *evaluating performance* and *evaluation of performance*. Compounds with different underlying roles will occur in different surface forms. Consider the compounds *computer science test* and *hearing test*, with underlying roles, *about* and *object*, respectively. The different roles account for the different surface forms: *test* **about** *computer science*, not *test* **of** *computer science* and *test* **of** *hearing*, not *test* **about** *hearing*. The phrase, *test* **about** *hearing* is an acceptable phrase, but the underlying relationship between the words is then an *object* relationship, not an *about* relationship. Different surface forms result, then, because the underlying roles are different. Thus, roles are important for interpreting compounds as well as predicting different surface forms for a compound.

1.3 Interpreting Compounds

Two types of nouns are important in interpreting compounds: *nominalizations* and *role-nominals*. A nominalization is a noun derived from a verb. For example, *performance* and *evaluation* are nominalizations of the verbs, *perform* and *evaluate*, respectively. It is very common in nominal compounds for one of the nouns to be a nominalized verb. To interpret these compounds, most often an element of the compound will fill a role of the verb that has been nominalized. Consider the following compounds:

- *shape description*: *shape* fills the **object** role of the verb, *describe*
- *spring skiing*: *spring* fills the **time** role of the verb, *ski*
- *Utah hiking*: *Utah* fills the **location** role of the verb, *hike*

The second type of noun is a role-nominal[2]. A role-nominal is a noun that is closely associated with some event. In fact, it fills the role of a verb associated with the role-nominal. For example, *food* is a role-nominal that fills the object role of the *eating* event, *performer* is a role-nominal that fills the actor role of the *perform* event, and *glasses* is a role-nominal that fills the instrument role of the *read* or *see* events. Knowing that a noun is a role-nominal also aids in the interpretation of nominal compounds. For example, given the compound *dog food*, (where *food* is a role-nominal that fills the object role of the *eating* event), it is possible to fill other roles of the *eating* event with other nouns in the compound. Namely, *dog* fills the actor role of the *eating* event and the interpretation for *dog*

food is: *food that dogs eat.* Thus, it is important to identify role-nominals and to know the events that are closely associated with them.

Determining the relationships between the words in compounds containing a nominalization or role-nominal is easier than in compounds containing strictly nouns. In the former type of compound, the relationship is usually one based on the roles associated with an underlying verb. Thus, in *shape description*, *shape* fills the *object* role of *describe* and in *dog food*, *dog* fills the *actor* role of *eat.* In strictly noun-noun compounds, like *computer routine* or *machine code*, the relationship is not as evident. In *computer routine*, the relationship could be one of the following: *designed-for, run-on, made-by*, or even a more generic relationship such as *for.* It is not easy to determine which of these relationships provides the best interpretation for the compound.

1.4 Information Retrieval

An application of interpreting nominal compounds is the field of Information Retrieval (IR), specifically Document Retrieval. In IR, a user poses a query to an IR system, such as:

Retrieve all documents on database management applications.

The system then looks through a set of documents and returns those documents that are related to the user's request. Most retrieval systems return documents based on the statistical frequency of words or their stems occurring in documents. Two problems occur with these systems: 1) many of the documents that are retrieved are not relevant and 2) many of the relevant documents are not retrieved. These two metrics help to rate the performance of IR systems. *Precision* measures the percentage of retrieved documents that are relevant, and *recall* measures the percentage of relevant documents that are retrieved.

The interpretation of nominal compounds could potentially improve the performance of IR systems in the following ways:

- Improve *precision* by determining that the concept expressed in a compound does not occur even though all words from a compound occur in the document. The words from *data type* occur in the phrase,

  all **types** *of* **data** *manipulation*,

  but the same concept as in the compound is not expressed. Interpreting compounds could help determine that these documents are irrelevant.

- Improve *recall* by recognizing the concept from a compound expressed in different surface forms. The compound, *performance evaluation*, occurs in text in the following forms:

  - **evaluating** *computer system* **performance**

The **performance** *of information retrieval systems can be* **evaluated**
— **evaluation** *work is based on measuring the retrieval* **performance**

Interpreting compounds could help the retrieval system recognize that documents containing these sentences are relevant documents.

- Improve *recall* by recognizing the concept from a compound using synonyms for the words in the compound. The concept, *optimization algorithm*, occurs in the following phrase:

  a simple **optimizing technique**

  Using the interpretation of compounds and synonyms for words in the compound, an IR system could recognize the concept from a compound in sentences like these.

- Improve *recall* by weighting the words in a compound by their importance and returning documents that contain the most important words. In the compound, *optimization algorithm*, *optimization* is the most important word. Retrieving a document with the phrase:

  object code **optimization**

  will undoubtedly be a relevant document.

Section 4 discusses results of an experiment conducted using the first two techniques above. The results show the possibility of improving the precision of IR systems by using the first technique. However, the second technique is not necessary because a simpler technique using an extension to a statistical retrieval method increases recall to almost 90%. The interpretation of compounds could not increase recall much beyond that level. Section 4 also discusses the techniques for using synonyms and weighting even though the experiment did not include these techniques.

1.5 Approach

The approach in this work is to use simple techniques and generalizations as much as possible because of the particular IR system for which this work was designed and the trade-off between the cost and benefit of incorporating natural language techniques in IR systems.

The work described in this report will be incorporated in the ADRENAL (Augmented Document REtrieval using NAtural Language processing)[3] system. The knowledge representation in ADRENAL is the REST (Representation for Science and Technology) language for general scientific and technological terms. While words like *experiment, data, analysis*, and *result* will be included in REST, domain specific words such as *computer, oscilloscope*, and *telescope* will not. Because of the lack of detailed knowledge about domain specific words, general

mechanisms for interpreting compounds are necessary. For example, even though the word *telescope* is not in REST, the system might be able to infer that *telescope analysis* means *analyzing with a telescope* if it knows that scientific instruments are used in analysis and it can determine that *telescope* is a scientific instrument. Thus, general mechanisms are necessary in order to interpret nominal compounds.

In IR, each new method introduced to improve precision or recall must be weighed against the amount of time and energy needed to incorporate this technique into an IR system. This is the trade-off between the cost and benefit of a technique. For example, a simple technique that improves precision and recall by a small amount might be more useful than elaborate techniques that improve performance by a larger amount simply because the time required to incorporate the elaborate techniques might not be worth the improvement in performance. The approach taken in this work was to look for simple techniques to interpret a large majority of compounds and to use these techniques to improve the precision and recall of IR systems.

1.6 Overview of this Report

The next section discusses previous work on nominal compounds including the approaches taken in studying compounds, the relationships between nouns in the compounds, and parsing compounds into constituent phrases.

Section 3 discusses the implementation for this work including the knowledge representation, the interpretation algorithm, and the results of the program interpreting two sets of nominal compounds.

Section 4 discusses the possibility of using this interpretation mechanism in information retrieval and gives some statistics that suggest that using this technique will improve the precision of information retrieval systems.

Finally, Section 5 gives the conclusions and discusses directions for future work.

2 PREVIOUS WORK

2.1 Approaches in Studying Nominal Compounds

Linguists and artificial intelligence researchers have studied nominal compounds for many years. Researchers initially focused on *generating* compounds from underlying clause structures using transformations. These underlying structures were first based on *syntactic* relationships. Later, researchers realized that the relationships between nouns in a compound were more *semantic* in nature. Gradually, the transformational approach to generating compounds was replaced by a computational approach to interpreting compounds.

Lees[4] believed that compounds were generated by applying transformations to sentential forms. The transformations could be applied to elements occurring in certain syntactic relationships within a sentence. For example, the *subject* and *predicate* could form a compound such as *fighter plane*. Similarly, the *subject* and *verb* could form a compound such as *talking machine*. In the 70's, Lees revised his approach so that the relationships between the elements of the underlying structures were more *semantic* in nature. The relationships were **case-frame** roles such as *agent, patient*, and *instrument* instead of **grammatical** roles, such as *subject, predicate*, and *object*.

Levi[5], in the 70's, had a similar approach to Lees' syntactic approach. However, she proposed that compounds were formed from underlying relative clauses or complement structures rather than from the types of syntactic structures proposed by Lees. In her approach, compounds were formed by two transformations: *deletion* and *predicate nominalization*. Deletion applies to a limited set of predicates called Recoverably Deleted Predicates (RDPs): *cause, have, make, use, be, in, for, from, about*. A compound is formed by applying a transformation to a relative clause containing one of these predicates. For example, the phrase *gas that causes tears* becomes *tear gas* and *top that a box has* becomes *box top*. In predicate nominalization, a verb is nominalized and either its subject or object becomes part of the resulting compound. Examples of compounds derived by this mechanism include: *dream analysis, enemy invasion*, and *student inventions*.

Rhyne[6] wrote a computer program to generate compounds from underlying relative clauses. His mechanisms for generating compounds were those proposed by Levi: deletion and predicate nominalization. The underlying relationships between the elements of a relative clause were case-frame roles such as: *performer, object, goal, source, location, means, cause*, and *enabler*, and not the RDPs as Levi proposed.

Downing[7] and Allen[8] criticized the transformational approach for the following reasons:

- The arbitrariness of the underlying structures. No principled reasons were given for the particular choice of Lees' syntactic structures or Levi's set of RDPs.
- Elements of a phrase that have semantic content were being deleted.
- The transformational rules and the classes of underlying structures did not seem to account for all types of compounds.

AI researchers in the late 70's and early 80's began looking at the task of using a computer to *interpret* nominal compounds. Finin[2] focused on determining the relationships between the words in compounds. His computer program contained a set of interpretation rules that gave an interpretation for a compound along with a score. From the different interpretations and scores, his program chose the best interpretation for a compound.

McDonald[9] also wrote a program to interpret compounds. Besides determining the relationships between words in a compound, McDonald looked at parsing compounds with more than two words and also in disambiguating multiple word senses. His approach incorporated more pragmatic information into the interpretation of compounds. To determine which interpretation was best for a particular compound he incorporated context and information about compounds his system had already interpreted.

Studying nominal compounds, then, changed from a transformational approach to generating compounds to a computational approach for interpreting compounds. The emphasis also shifted from the *syntactic structure* of compounds, to the realization that the *semantics* of the nouns in a compound contributed to its interpretation.

2.2 Relationships Between Nouns in a Compound

Researchers proposed different mechanisms to account for the relationships between nouns in a compound. Initially, they proposed *fixed* lists of relationships. In later years, they used knowledge about words and the relationships in which they are likely to occur to account for the different relationships.

Lees[4] derived an elaborate taxonomy of the types of *syntactic* relationships underlying compounds. The main types were:

subject-predicate: *girl friend, fighter plane*
subject - middle object: *arrow head, rattlesnake*
subject - verb: *talking machine, population growth*
subject - object: *steam boat, car thief*
verb - object: *pick pocket, eating apple*
subject - prepositional object: *gunpowder, garden party*
verb - prepositional object: *washing machine, boiling point*

object - prepositional object: *wood alcohol, block house*
proper nouns and naming: *ferris wheel, Marshal Plan*

Each of these main categories was divided into more detailed categories.

Levi also proposed a fixed list of relationships, her recoverably deleted predicates[5]: *cause, have, make, use, be, in, for, from, about.*

Li[10] proposed 24 compounding relationships, including *made of, made from, made with, used for, are parallel.* Li admitted his 24 relationships did not account for the generation of *all* compounds. He defined a class of compounds that were the product of no compounding mechanism, such as **cradle song**: *a song to lull a child in the cradle to sleep* and **dishwater**: *water in which one has washed dishes.*

All these lists of relationships were criticized for many reasons, the main reason being that no underlying criteria existed for choosing the particular relationships in the lists. Thus, Levi proposed nine RDPs, but her list could just as well have contained more or fewer RDPs to account for the same phenomenon. Furthermore, Li proposed 24 compounding relationships and a group of non-compounding relationships, but he gives no reason why some relationships fall in one category and not the other. Also, these lists could not account for all the types of relationships in compounds and sometimes the relationships in a compound could be described by several elements in the list. For example, several RDPs could be used as the underlying relationship for the compound, *box top.* The relationship could either be *for* or *have*: *top* **for** *a box*, or *top that a box* **has**.

Researchers began to incorporate semantic and pragmatic knowledge in order to determine the relationships between nouns in a compound. Downing[7] suggested the following information to determine the relationships:

- the types of relationships in which individual nouns can occur,
- the context in which a compound is used, and
- the semantic class of the nouns in the compound.

In Rhyne's[6] program, a compound can be formed only if a characteristic aspect is associated with one of the nominal elements in the compound. Thus, if a *boat* is characteristically used to catch *turtles*, then the boat may be called a *turtle boat.* Allen[8] believed the relationship between the nouns in a compound could be specified in terms of the interaction of the *semantic feature* hierarchies of the two nouns in the compound. To form a compound, the semantic content of the modifier of the compound "plugs into" any of the semantic feature slots of the head. The "best" relation is determined by the *dominant* semantic features of the head. For example, the compound *shoe box* most likely means a *box for containing shoes* because *container* is a dominant semantic feature of *box.*

In determining the relationships between nouns in a compound, the focus shifted from fixed lists of relationships to knowledge about the nouns in a

compound including: 1) the salient features of nouns, 2), their semantic class, 3) the activities in which they characteristically occur, and 4) the context in which they are used.

2.3 Parsing Nominal Compounds into Constituent Phrases

Parsing nominal compounds into constituents is an area that researchers have not studied in depth. The ideas discussed here show the importance of identifying nominalizations and role-nominals in compounds and are a starting point for further research in this area.

McDonald[9] in his thesis, lists four patterns of nominal compounds:

- Compounds that can be parsed in a left-to-right manner, i.e. [[[N_1 N_2] N_3] ... N_n]. Examples: *science teacher institute, coal mine supervisor, blood donor recruiter*, and *sugar cane plantation owner.*
- Compounds that can be parsed in a right-to-left manner, i.e. [N_1 [N_2 .... [N_{n-1} N_n]]]. Examples: *glass wine glass, liquid roach poison*, and *network operating systems.*
- Compounds with a nominalized verb in the middle, i.e. [N_1 V N_2] where N_1 and N_2 are nouns or nominal compounds and V is a nominalized verb. Examples: *car assembly plant, food distribution center, watch repair person*, and *[water meter cover] adjustment screw.*
- Compounds with a nominalized verb at the end, i.e. [N_1 N_2 V] where N_1 and N_2 are nouns or nominal compounds and V is a nominalized verb. Examples: *student course evaluation, government price support, executive stock purchase*, and *January [automobile water pump cover] shipments.*

With these last two patterns, McDonald has shown the importance of recognizing a nominalization in a compound. Having recognized a nominalization, it is possible to check words preceding or following the nominalization to see if the words could fill roles of the compound. This seems to be a promising starting point in parsing compounds.

Isabelle[11] gives more information on parsing compounds containing nominalizations. In compounds that have a nominalized verb or role-nominal at the head, Isabelle[11] gives the following ordering for the modifiers:

```
time, location  <  paths  <  subject  <  object
```

Thus, if a compound with a nominalized verb or role-nominal at the head has a *location, subject*, and *object* modifier, then the modifiers would appear in the following order: [*location subject object* head]. The following compounds satisfy this ordering constraint:

computer fuel testing	?? fuel computer testing
Montreal jet flights	?? jet Montreal flights
January [automobile water pump cover] shipments	?? [automobile water pump cover] January shipments

Thus, identifying a nominalization or a role-nominal is important in parsing compounds into constituents.

3 IMPLEMENTATION

The focus of the implementation was to determine some simple techniques to interpret the majority of nominal compounds because of the desire to incorporate simple natural language techniques in the ADRENAL information retrieval system.

The implementation had three stages:

- Designing the knowledge base
- Writing the interpretation algorithm, and
- Testing the algorithm on two sets of nominal compounds.

The following sections discuss these stages in more detail.

3.1 Knowledge Representation

Knowledge representation has been one of the most important parts of this research because knowledge about concepts and the relationships in which they participate is essential in interpreting compounds. Knowledge about the world enables us to interpret compounds, and to know, for example, that a *milkman* is someone who *delivers* milk while a *garbage man* is someone who *takes away* the garbage.

The knowledge base designed for this work uses features that any knowledge base must have if it is used for interpreting compounds. These features are:

- concepts that correspond to individual word senses,
- categories of concepts for making generalizations about groups of words,
- roles for concepts that indicate in which relationships the concepts are likely to occur,
- semantic preferences for roles to indicate the types of objects that may fill the roles,
- salient roles that indicate in which relationships a concept is more likely to occur, and
- role-nominals and nominalizations to facilitate the interpretation of compounds.

Sections 3.1.1 - 3.1.4 describe these features in more detail and Section 3.1.5 describes the implementation details.

3.1.1 Concept dependent approach

The interpretation of nominal compounds is heavily dependent upon the concepts in the compound; the interpretation is concept *dependent*, not concept *independent* as Finin[2] points out. A concept *independent* approach is like that of

Lees and Levi who proposed a fixed list of the relationships possible in nominal compounds. Interpreting a compound using this approach involves scanning the list of relationships and determining which one best describes the relationship in a compound. Downing[7] and Allen[8] pointed out the inadequacy of these lists.

The concept *dependent* approach involves using knowledge about individual concepts to determine the relationships between nouns in a compound. Consider the following three compounds: *milkman*, *garbage man* and *mail man*. To understand the differences between these compounds, knowledge about the concepts *milk*, *garbage*, and *mail* are used:

- *Milkman* is someone who *delivers* milk because *milk* is able to be delivered,
- *garbage man* is someone who *takes away* garbage because *garbage* is something people throw away, and
- *mail man* is someone who *delivers* **and** *takes away* mail because *mail* is something that people both send and receive.

The concept dependent approach means that knowledge about the concepts must be able to suggest relationships in which the concepts can occur.

3.1.2 Categories

To encode knowledge about concepts, the concepts in the knowledge base are organized into categories. This division defines groups of words that behave in a similar manner and allows generalizations to be made about the concepts in a particular category. For example, when the words *technique* and *mechanism* occur in compounds such as

> *analysis technique, analysis mechanism, evaluation technique, evaluation mechanism,*

they are acting as the *instruments* of the nominalized event that is the modifier of the compound. These compounds show that the words *technique* and *mechanism* are interpreted similarly in compounds that have a nominalized verb as the modifier. To facilitate the interpretation of similar compounds, the following information must be encoded in the knowledge base:

- *technique* and *mechanism* (and similar words) form a category, and
- the *instrument-of* role becomes a role for that category.

Dividing the knowledge base into categories is also motivated by the IR system, ADRENAL, and the REST knowledge representation. In this domain many of the compounds encountered will contain words not in the REST hierarchy since REST only contains general scientific terms. By identifying the REST category to which these unknown words belong, it could be possible to interpret a compound with unknown words. Consider the compound, *addressing*

scheme. If *scheme* is not in the hierarchy, but is identified as belonging to the same category as the words *technique* and *mechanism*, then the system might propose that the relationship in this compound is the *instrument-of* relationship. Thus, it might be possible to interpret compounds even if one or both of the words in the compounds are not included in the REST hierarchy.

Studying several hundred nominal compounds occurring in the CACM document collection** helped to define categories in the knowledge base and to make generalizations about those categories. Figure 1 shows some of those categories and generalizations. It also lists compounds using words from the category that exemplify the generalizations.

3.1.3 Roles

The roles for each concept in the knowledge base indicate the relationships in which that concept can occur. Each role has semantic preferences that limit the type of concepts that may occur in each relationship, and salient roles indicate the relationship in which a concept is most likely to occur.

Each concept in the knowledge base has roles associated with it. For example, the *analysis* concept has *agent, object, instrument, location*, and *time* roles, and *procedure* has an *instrument-of* role. As Downing pointed out, relationships between words in nominal compounds are based on permanent characteristics or functions of the words. These roles serve that purpose. Thus, in identifying that *features, capabilities*, and *characteristics* all act in a similar manner when they are the head of a compound, the *feature-of* role is added to these concepts. Similarly for the *instrument* words described above; a characteristic of these words is that they act as *instruments of* events, so they have an *instrument-of* role.

The roles have semantic preferences (or selectional restrictions) that serve to limit the types of objects that may fill a role. A typical preference is that the actor of an event must be *animate*. These preferences are *categories* (*animate* is a category in the hierarchy) so they indicate the categories of concepts that may fill the roles.

The salient roles for a concept facilitate the interpretation of nominal compounds. By identifying salient roles, the interpretation algorithm can determine which role is more likely to describe the relationship between nouns in a nominal compound. For example, in the corpus of several hundred compounds, the modifiers of compounds that had a nominalization at the head were the *objects* of the nominalization, as in *data analysis* and *pattern recognition*. Thus,

**The set of documents and queries used in this work is from a standard test collection that the IR community uses to compare the performance of different IR systems. The document set contains 3204 articles published in the *Communications of the ACM* from 1959 to 1979 and the query set contains 64 queries on computer science topics[12].

INSTRUMENT CATEGORY

Generalization: Compound form: *nominalization instrument*
Interpretation: the head is the *instrument-of* the *nominalization*

Words in Category	Examples of Compounds
mechanism	*control mechanism*
program	*simulation program*
technique	*evaluation technique*
method	*accessing method*

TRANSITIVE EVENT CATEGORY

Generalization: Compound form: *x transitive-event*
Interpretation: *x* is the *object* of the *transitive-event*

Words in Category	Examples of Compounds
analysis	*data analysis*
management	*database management*
construction	*dictionary construction*
design	*systems design*

FEATURE CATEGORY

Generalization: Compound form: *x features*
Interpretation: the head is the *features-of x*

Words in Category	Examples of Compounds
features	*system features*
capabilities	*computing capabilities*
characteristics	*performance characteristics*

Figure 1: Categories of Words

the *object* role is a salient role for concepts such as *analysis* and *recognition.* Identifying salient roles is similar to the approach taken by Allen, who listed the features for a word in a hierarchy, with the *dominant* features determining which relation is best in a particular compound.

3.1.4 Role-nominals and nominalizations

The knowledge base must also identify role-nominals and nominalizations. Role-nominals must be identified so compounds like *dog food* can be interpreted as *food that dogs eat.* The *cat* concept is important in the interpretation, but does not occur explicitly in the compound; rather, it is implicitly associated with *food.* The knowledge base makes explicit the relationship between *food* and *eat.*

Nominalizations must be identified so that they inherit the verbal roles of the verb from which they were derived. Thus, words such as *analysis, performance,* and *evaluation* must inherit the roles associated with the verbs *analyze, perform,* and *evaluate,* respectively.

Both of these types of nouns must inherit verbal as well as nominal roles. Consider the compound, *aluminum water pump* with the interpretation: *a pump that pumps water and is made of aluminum. Pump* is a role-nominal in this compound; it fills the *instrument* role of the *pump* event. *Pump's* relationship with *water* is an *object* relationship (a verbal role), while the relationship with *aluminum* is a *made-of* relationship (a nominal role). Thus, both verbal and nominal roles are necessary to interpret this compound. Furthermore, if interpreting compounds can be generalized to noun-modification in general, then the ability to associate verbal and nominal roles with a nominalization is essential in interpreting phrases such as *large data analysis* where *large* is the *size* of the *analysis* (a nominal role) and *data* is the *object* of the *analysis* (a verbal role).

3.1.5 Implementation details

The knowledge representation used in this work is an extension of the Ace knowledge representation developed by Paul Jacobs[13]. Ace is a descendant of KODIAK[14], a hierarchical frame-based representation language. Ace was extended so that the roles could have semantic preferences.

The knowledge base contained approximately 180 concepts and 38 categories. The difference between concepts and categories is that the concepts correspond to words while the categories do not. Most of the concepts encoded in the knowledge base were from the words found in the compounds in the CACM queries.

The knowledge base is divided into three main categories: *events, entities,* and *relations.* Each category had several subdivisions with some categories derived from the study of the compounds from the CACM collection and some categories

derived from a thesaurus. Table 1, given earlier, lists some of the categories. The knowledge base contained approximately 60 events, 95 entities, and 25 relations.

An example of a category from the hierarchy is given below.

```
(c-ent c-construction
       (PAR c-transitive-event)
       (PR (thing-constructed (c-entity))))
```

This category is the *c-construction* category. Its parent is the *c-transitive-event.* This category only has only role, the *thing-constructed* role. The semantic preferences on this role are that the *thing-constructed* be an *entity.*

Most of the semantic preferences on the roles are very general, such as the *object* of a *construction* event has to be an *entity.* These restrictions are so general because the hierarchy does not have many categories. However, some preferences are specific, such as the preference that *actors* of events be *animate.* Furthermore, the preferences on *location* and *time* roles are that the role be a *location* or a *time* entity.

The knowledge base did not include a lexicon because each word corresponded to a single concept in the knowledge base; multiple word senses were not allowed. This constraint was necessary to reduce the complexity of the interpretation process. However, the roles and semantic preferences might help in disambiguating multiple word senses if the knowledge base included them because the roles define possible relationships in which the words might occur. For example, the word *matching* has many word senses, but in *matching funds* the particular word sense is **providing** *funds* while in *pattern matching* the sense is **finding similar** *patterns.* The restrictions on the *object* role for *matching* might aid in determining these different word senses.

Each concept in the knowledge base corresponds to a single word; not a phrase. Incorporating compounds as phrases in the lexicon is not possible because of the REST representation language that contains general scientific terms. REST is not designed to include all possible words or compounds that might occur in the document collections; it is designed for general scientific concepts.

3.2 Interpretation Algorithm

The interpretation algorithm is designed to produce interpretations for a large majority of nominal compounds. Most compounds can be interpreted by the modifying noun filling a role of the head noun, as McDonald[9] points out. Thus, *code* fills the *object* role of *compact* in *code compaction* and *carpet* fills the *texture-of* role of *texture* in *carpet texture.*

Because the majority of nominal compounds contain only two or three nouns, the interpretation algorithm only interprets compounds with two nouns. In three

Table 1: Number of Words in Three Groups of Nominal Compounds

	CACM Documents		CACM Queries		McDonald's Thesis	
2-word NCs	146	72%	70	78%	523	84%
3-word NCs	49	24%	17	20%	94	15%
4-word NCs	8	4%	2	2%	7	1%
6-word NCs					1	<1%
Total:	203	100%	89	100%	625	100%

separate groups of nominal compounds, more than 95% of the compounds contained two or three words. These sources were the following:

- A random sampling of 203 nominal compounds from documents from the CACM collection
- The 89 compounds from the CACM query set
- The 625 compounds listed in David McDonald's thesis[9]

These data are listed in Table 1.

Being able to interpret compounds with two nouns could easily be extended to interpret compounds with three nouns. Since there are only two parses for compounds with three nouns, the interpretation mechanism could produce an interpretation for both parses and then decide which parse gave the best interpretation.

The interpretation algorithm is as follows:

1. If the head of the compound is a role-nominal, reset the head to be the event associated with the role-nominal and fill the appropriate role with the role-nominal
2. Find the roles of the head noun for which the modifier satisfies the semantic preferences
3. If more than one role is returned:
   (a) Rule out roles that are already filled
   (b) Favor the salient roles of the head noun
4. Return the role(s) found

The first step in the algorithm is an easy way to access the event associated with a role-nominal and is simply a bookkeeping step. The second step gathers all the

roles that the modifying concept could fill in the head noun making sure the modifying concept satisfies the semantic preferences for those roles. If that step returns more than one role, steps 3a and 3b help to determine which role is the correct role.

Step 3a was not implemented, but would be used to rule out roles that had already been filled by role-nominals or by other elements in a sentence (when context is incorporated). For example, *computer analysis* has two possible interpretations:

> *analysis* **of** *a computer* (*computer* fills *object* role of *analysis*) or
> *analysis* **by** *a computer* (*computer* fills *instrument* or *agent* role of *analysis*)

However, if the compound occurs in the phrase: *computer analysis of data*, then the only possible interpretation for the compound is the one where *computer* is the *instrument* or *agent* of the *analysis* event, because *data* fills the *object* role. McDonald[9] used context in this manner to rule out possible interpretations.

Step 3b favors the salient roles of the head noun. The salient roles were defined to be those roles that denoted permanent characteristics of a concept, so this step in the algorithm favors those roles that a person would be most likely to use when forming a relationship between words in a compound.

The output of this algorithm is a list of possible roles the modifying noun could fill in the head noun.

3.3 Results

This section describes the performance of the interpretation algorithm on two sets of nominal compounds. The first set of compounds were the compounds used to encode the knowledge base while the second set were compounds collected *after* the knowledge base was designed. The purpose of the second set of compounds is to test the algorithm on compounds for which the knowledge base was not designed.

The results show the difference between interpreting compounds containing a nominalization or role-nominal compared to those that do not. For instance, the relationship in *data analysis* is the *object* relationship; a role of the verb, *analyze*. In *computer routine*, however, the relationship could be one of the following: *designed-for, run-on, made-by*, or even a more generic relationship such as *for*. Determining the appropriate relationship in this compound is not as simple as in the compound, *data analysis*. The results of these two tests reflect this fact and show the need for more detailed analysis of the types of relationships occurring between nouns in strictly noun-noun compounds.

3.3.1 First test

The first group of compounds interpreted were the 87 nominal compounds used to encode the knowledge base. Some typical interpretations are:

```
Interpreting:  ACCESSING METHOD
Possible roles: (INSTRUMENT-OF)

Interpreting:  BIT STREAM
Possible roles: (STREAM-OF)

Interpreting:  CODE OPTIMIZATION
Possible roles: (THING-OPTIMIZED)

Interpreting:  CODE COMPACTION
Possible roles: (THING-COMPACTED)

Interpreting:  DATABASE PACKAGE
Possible roles: (TYPE)

Interpreting:  STAR HEIGHT
Possible roles: (HEIGHT-OF)
```

For 82% (71) of the compounds interpreted, the program produced the correct interpretation. "Correct interpretation" means the output of the program was one role and that role accurately described the relationship between the words in the compound. The interpretation of the six compounds listed above are correct interpretations. For the other 18% (16) of the compounds, there were many different problems, including:

- Strictly noun-noun compounds:
  - **human interface:** *(part-of has-part)*
  - **machine code:** *(has-part part-of)*
  - **relation matrix:** *nil*

  Determining the relationships in these compounds is difficult. Studying other compounds containing the nouns *interface*, *code*, and *matrix* would help to identify salient roles for these nouns.
- More than one salient role:
  - **computer performance:** *(instrument affects actor)*
  - **systems performance:** *(instrument affects actor)*

  The correct relationship in the above cases is either the *actor* or the *instrument* role, but the program returned the roles listed above because it

had no way to determine which role best described the relationship. Perhaps the salient roles for *performance* should have a preference for one of these three roles.

- Wrong semantic preferences:
  - **index selection:** *(goal goal-of affects affected-by causes caused-by)*

  The correct relationship for *index selection* is the *object* relationship: *selecting an index.* The semantic preference on the object role for *selection* is that the object be an entity. However, *index* is an event in the knowledge base so it is not allowed to fill the *object* role.

- Complex relationship:
  - **English spelling:** *(object)*

  With this compound, the role returned does not adequately describe the relationship. The correct interpretation is not, *spelling English*, rather it is *spelling using the English language* or *spelling words in English.*

For the first test, the words were encoded in the knowledge base specifically so the interpretation algorithm would produce the correct interpretation. Some of the interpretations failed only because the correct roles had not been encoded in the knowledge base. Thus, the performance of the algorithm could potentially be increased to almost 100%. However, the purpose of designing the knowledge base and algorithm was not to hard-code all the "right" relationships in the knowledge base, but rather to design a general mechanism for interpreting compounds. Testing the algorithm on a second set of compounds, then, was necessary to identify its general performance.

3.3.2 Second test

The second test of the nominal compound interpreter involved extracting compounds from the CACM document collection that used words already encoded in the knowledge base. A set of 131 compounds were used in this second test. Of these compounds, only 65% (85) produced the correct interpretation while 35% (46) were incorrect. The compounds with incorrect interpretations fall into the following groups:

- Strictly noun-noun compounds:
  - **computer memory:** *(part-of has-part)*
  - **file system:** *(type)*
  - **data security:** *(part-of has-part)*

  Thirty-one of the compounds were of this type. The roles returned do not adequately describe the relationship.

- More than one salient role:
  - **computer operation:** *(instrument affects actor)*
  - **systems operations:** *(instrument affects actor)*

  The correct relationship for these compounds is either the *actor* or the *instrument* role, but the program returned the roles listed above because it had no way to determine which role best described the relationship.
- Correct role not in the knowledge base
  - **design decision:** *(goal goal-of affects affected-by causes caused-by)*
  - **programming decision:** *(goal goal-of affects affected-by causes caused-by)*

  For these compounds, the correct role was not encoded in the knowledge base. The correct role is an *about* relationship.
- Wrong semantic preferences:
  - **index construction:** *(goal goal-of affects affected-by causes caused-by)*
  - **design characteristics:** *(affected-by)*
  - **search capabilities:** *(affected-by)*

  In these compounds, the correct relationship is not returned because of the semantic preferences. In *index construction* the correct relationship is the *object* relationship. The semantic preferences on the *object* role for *construction* is that the object be an **entity**. However, *index* is an **event** in the knowledge base so it is not allowed to fill the *object* role. The same problem occurs with *design characteristics* and *search capabilities*; the correct role is the *characteristics-of* or *capabilities-of* role. The semantic preferences on these roles are for **entities**, not **events**, but *design* and *search* are encoded as **events** in the hierarchy.
- Role-nominal not encoded:
  - **computer user:** *(part-of has-part)*
  - **language user:** *(part-of has-part)*

  The interpretation for these compounds is that the *user* is *using* a *computer* (or *language*): *use* is the event associated with *user*, and *computer* and *language* fill the *object* role of this event. However, *user* was not encoded as a role-nominal in the knowledge base, so the algorithm did not return the correct role.

All problems but the first are problems with the knowledge base: the correct roles, the salient roles, the semantic preferences, and role-nominals were not encoded correctly for all the concepts in the knowledge base. By solving these problems,

the performance of the algorithm on the second test could be increased to 76%. The performance could be increased even more by determining the types of relationships that occur in strictly noun-noun compounds.

These tests show that the interpretation algorithm is able to interpret a majority of compounds using a knowledge intensive approach. However, the performance is lower when interpreting strictly noun-noun compounds and compounds containing words whose roles or semantic preferences were not encoded correctly in the knowledge base. To increase performance, the following is necessary: 1) a more detailed analysis of the types of relationships occurring between nouns in strictly noun-noun compounds and 2) the addition of more knowledge about the concepts in the knowledge base.

4 INFORMATION RETRIEVAL

An experiment was conducted to determine how the interpretation of nominal compounds might improve the precision and recall of an IR system. Specifically, to improve precision by recognizing that the same concept from a compound is not expressed even though all the words from a compound occur in the same sentence in a document, and to improve recall by recognizing the concept from a compound in different surface forms within a single sentence.

The next sections discuss the motivation for the experiment, the results of the experiment, and two other techniques that could potentially improve the recall of IR systems.

4.1 Motivation

Interpreting compounds could improve the performance of IR systems because compounds occur frequently in user queries and because the concepts that nominal compounds refer to occur in text in many different surface forms. In the set of 64 computer science queries for documents from the CACM, 89 nominal compounds occurred. Thus, on the average, at least one nominal compound appeared in every query. The following query contains five nominal compounds:

> *I'd like papers on design and implementation of* **editing interfaces, window-managers, command interpreters,** *etc. The essential issues are* **human interface design,** *with views on improvements to* **user efficiency,** *effectiveness and satisfaction.*

This particular query demonstrates how often nominal compounds are used in IR queries, especially in queries for documents from a technical field.

Compounds occur in many different surface forms in text. Consider the compound, *performance evaluation.* In abstracts from 20 documents from CACM, this concept occurred in the following forms when both words from the compound occurred in the same sentence:

- *performance evaluation*
- *performance evaluations*
- *Performance Evaluator*
- *evaluating performance*
- *evaluating* the *performance*
- *evaluating* computer system *performance*
- the *performance* of a proposed design is not *evaluated*
- The *performance* of information retrieval systems can be *evaluated*
- *evaluation* work is based on measuring the retrieval *performance*

- *evaluation* is based on optimizing the *performance*

The interpretation of compounds might improve recall of IR systems by being able to recognize all these different forms for this compound. Recognizing these forms would involve parsing the sentences above and then comparing this output to the interpretation of the compound given by the interpretation mechanism in Section 3.2. For example, the interpretation mechanism returns the *object* role as the relationship in this compound. In all but the last two phrases above, the relationship between *performance* and *evaluate* is also the *object* relationship. Thus, a program might be able identify these sentences as containing the same concept as expressed in the compound.

Interpreting nominal compounds should also allow an IR system to determine that the concept expressed in a nominal compound does not occur in a document even though all the words from a compound occur. This mechanism would improve the precision of IR systems by weeding out those documents not containing the concept from the compound. For example, the interpretation mechanism gives the interpretation *structure for data* for *data structure*, but the interpretation in the phrase

structure *of* **data**

is the *organization of data.* In the compound *data type*, the interpretation is *something that determines the type of data* while in the phrase,

transmitting **data** *of a* **type**,

the relationship is *data of a particular type.* Furthermore, the relationship in *programming language* is *used-for*, but in the phrase,

programs *written in an ALGOL-like* **language**,

the relationship is *written-in.* Thus, a program might be able to determine that these phrases did not express the concept in the compound even though the words from the compound occurred in the phrase.

Standard IR systems retrieve documents based on the statistical frequency of the words in a query appearing in the text. The interpretation of nominal compounds might improve performance for these systems, but only for those documents that contained the words from the compound *in the same sentence.* Using current natural language techniques, a system could recognize a concept from a nominal compound if it is expressed in a single sentence, but if the concept is expressed across sentences, the system would not be as likely to recognize the concept. Consider, the following text:

- *A method is presented for* **evaluating** *computer system* **performance** *in terms of a cost/utilization factor and a measure of imbalance.*

- *A study comparing the* **performance** *of several computer programs for integrating systems of ordinary differential equations is reported. The integration methods represented include multistep methods (predictor-correctors), single-step methods (Runge-Kutta) and extrapolation methods (both polynomial and rational). The testing procedure is described together with the* **evaluation** *criteria applied.*

In the first example, the words from the compound *performance evaluation* appear in the same sentence, while in the second example the compound appears across sentences. A natural language system would be more likely to recognize the concept in the first example than in the second. Although NL techniques such as discourse analysis and anaphora resolution are able to link concepts occurring across sentences, the data presented here only show the possibility of recognizing the concept from a compound when the words from the compound occur *in the same sentence.*

4.2 Experiment

The experiment involved analyzing documents retrieved from a standard statistical retrieval system, the I^{3}R[15] system developed at the University of Massachusetts. The system retrieved the 20 highest-ranked documents for the 17 queries listed below.

bit stream	*image processing*
computer graphics	*optimization algorithms*
computing structure	*pattern recognition*
control mechanism	*performance evaluation*
data structure	*procedure calls*
data type	*programming languages*
disk head	*space efficiency*
file handling	*user efficiency*
graph isomorphism	

The compounds used as queries are a subset of the 89 compounds found in the CACM query set. The text retrieved by the system was the abstract from each of the 20 documents.

The retrieval method used a stemming algorithm, so the retrieved documents contained different morphological forms of the words in the queries. The following morphological variants of *evaluation* appeared in the abstracts retrieved for the query, *performance evaluation*:

evaluating, evaluated, evaluations, evaluator, and *evaluate.*

The morphological variants for the word *performance* using the same query were:

perform and *performs.*

In the abstracts returned, 186 sentences contained both words from the compound used as the query for that particular document. Analysis of the 186 sentences showed that in 66% (122) of the sentences, the concept expressed in the compound *was* in the sentence, and in 31% (58) of the sentences, the concept *was not* in the sentence. In 3% (6) of the sentences, the concept was expressed using words *other than* the words in the compound even though both words from the compound occurred in the same sentence. These data are shown in Table 2.

Table 2: Sentences Containing Both Words from a Compound

Same concept using same words	122	66%
Different concept using same words	58	31%
Same concept using different words	6	3%
Total:	186	100%

An example of a sentence from each category is listed below. These sentences contain words from the compound, *procedure call.*

Same concept using same words: *String* **procedures** *may be declared and* **called** *in a standard ALGOL context.*

Different concept using same words: *This integration is accomplished by treating* **procedures** *and their activation records (*called *environments) as data objects and by decomposing procedure invocation into three separate components at the source-language level.*

Same concept using different words: *This integration is accomplished by treating procedures and their activation records (called environments) as data objects and by decomposing* **procedure invocation** *into three separate components at the source-language level.*

Thirty-one percent of the sentences, then, contained both words from a compound, but these words did not express the same concept as in the compound. This provides preliminary evidence that interpreting nominal compounds could improve the precision of an IR system by distinguishing between the sentences that contain the same concept as in a compound, and those that do not.

An alternative to using the interpretation of nominal compounds might be to extend one of the current statistical methods in order to improve precision.

Specifically, an enhanced version of one of these methods might reject the cases where the two words in a compound appear in the same sentence, but they do not express the same concept as in the compound. A simple extension is to accept those documents that contain a sentence where the two words from a compound are adjacent. A large majority of the sentences are actually of this type. Figure 2 shows the performance of this new technique.

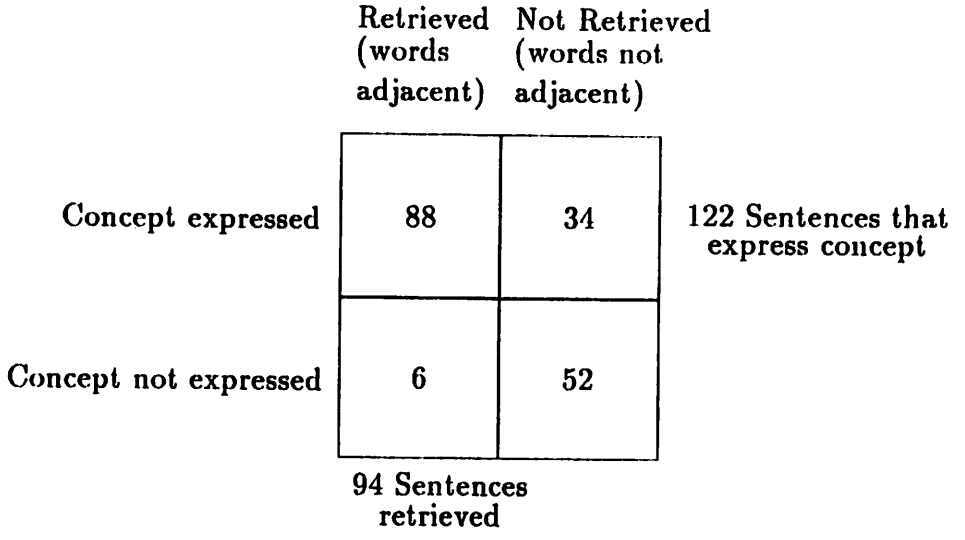

Figure 2: Breakdown of Technique for Adjacent Words

This technique returns 94 sentences - in these sentences the two words from the compound are adjacent to each other. Examples of sentences returned by this technique that do and do not contain the concept in a compound are given below:

Words adjacent and concept expressed: *This paper presents a notation and formalism for describing the semantics of* **data structures.**

Words adjacent and concept not expressed: *Language Problems Posed by Heavily* **Structured Data**

This technique had 94% precision: of the 94 sentences retrieved, 94% (88 out of 94) of the sentences contain the concept from the compound and only 6% (6 out of 94) of the sentences do not contain the concept from the compound. The recall for this technique is not as high, however. Of the 122 sentences that contained the concept as expressed in the compound, only 72% (88 out of 122) of these sentences were actually retrieved (the words from the compound were adjacent) while 28% (34 out of 122) were not retrieved (the words from the compound were not

adjacent). In summary, the performance of this technique is 94% precision and 72% recall.

Consider another extension to a statistical retrieval method. This technique returns those sentences where the words from the compound occur within three words of each other instead of occurring adjacent to each other. Figure 3 shows the breakdown for sentences processed using this new technique. This new

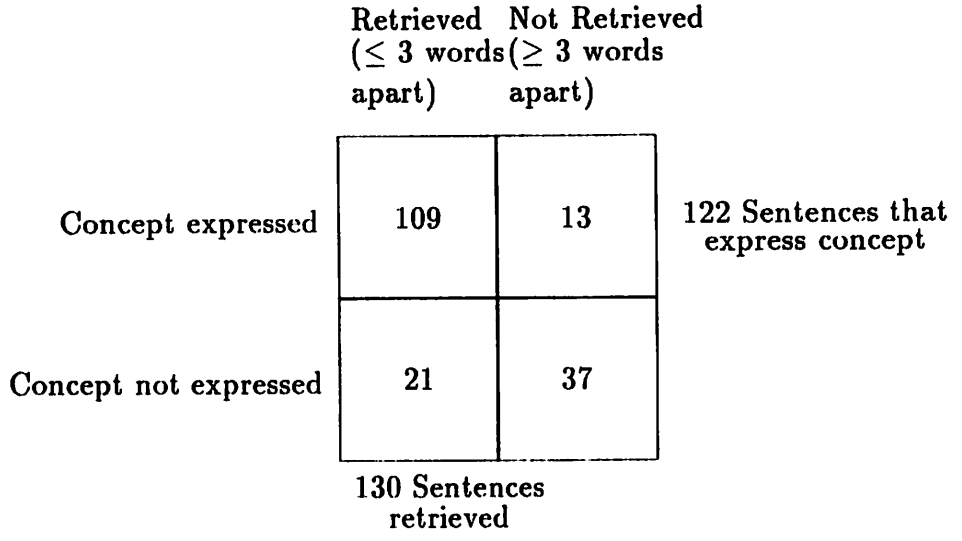

Figure 3: Breakdown of Technique for ≤ Three Words Apart

technique has a precision of 84% and a recall of 89%. For precision, 84% (109 out of 130) of the retrieved sentences expressed the concept while 16% (21 out of 130) did not express the concept. For recall, 89% (109 out of 122) of the sentences that expressed the concept were retrieved while 11% (13 out of 122) were not retrieved. Compared to the previous technique (retrieving sentences where the two words from a compound are adjacent), this technique has lower precision, but higher recall. Table 3 shows the comparison of these two techniques.

This second technique (retrieving those sentences where the words from the compound occur within three words of each other) has a very high recall rate. The interpretation of nominal compounds could not improve this recall rate much without hurting precision, but perhaps it could improve the precision of the second technique while maintaining the high recall. To increase the precision, the system would weed out documents that have sentences that contain the words from a compound within three words of each other, but that do not express the

Table 3: Comparison of Extensions to Statistical Retrieval Methods

	Precision	Recall
Words adjacent in sentence:	94%	72%
Words $\leq$ 3 words apart in sentence:	84%	89%

same concept in the compound. Examples of such sentences are:

- **(computer graphics):** *An Interactive* **Computer** *System Using* **Graphical** *Flowchart Input*
- **(data structure):** *The implications of these features for* **data** *layout and algorithm* **structure** *are discussed.*
- **(data type):** *A method has evolved for transmitting* **data** *of a* **type** *originating in many laboratory situation direct to a central computer.*
- **(file handling):** *Techniques developed for* **handling** *scroll editing,* **filing,** *and the layered system structure are outlined.*
- **(user efficiency):** *Two languages enabling their* **users** *to estimate the* **efficiency** *of computer programs are presented.*

By recognizing the underlying roles between the words in the compound, and parsing the sentences above, a system could recognize that the same concept was not being expressed. For example, parsing the last sentence above shows that it mentions the *efficiency of computer programs*, not the *efficiency of a user*, so the concept in the compound is not expressed in that sentence. Using the interpretation of compounds in this manner could potentially improve the precision of the second technique by a maximum of 16%.

4.3 Other Techniques

Other areas where the interpretation of nominal compounds could improve IR techniques is in documents that contain the same concept from a compound, but that use synonyms for one or both of the words in the compound. This technique could improve recall by returning those documents using synonyms for the words in a compound. Examples of such sentences are the following:

- **(data structure):** *A variety of* **data representations** *that have been used to describe structured information are then examined, and the characteristics of various processing languages are outlined in the light of the procedures requiring implementation.*

- **(procedure call):** *This integration is accomplished by treating procedures and their activation records (called environments) as data objects and by decomposing* **procedure invocation** *into three separate components at the source-language level*
- **(optimization algorithm):** *Redundant instructions may be discarded during the final stage of compilation by using a simple* **optimizing technique** *called peephole optimization.*

Another way to increase recall would be to use the interpretation of compounds to weight the words in a statistical retrieval mechanism. In some compounds, one of the words is more important than the other. To reflect this fact, the system could assign a higher weight to the more important word. In the compound, *optimization algorithm*, *optimization* is the most important concept in this compound. Retrieving a document with the phrase *object code optimization* will undoubtedly be a relevant document.

5 CONCLUSIONS

This section discusses areas in which the research described in this report could be extended and then sums up the accomplishments of the research.

5.1 Future Work

This report did not address polysemy, but it is an important part of any natural language or information retrieval system. To disambiguate multiple word senses the system would need a lexicon and a mechanism for mapping words in the lexicon to concepts in the knowledge base. The roles in the knowledge base might help in mapping to a single word sense, because they indicate the relationships in which each concept (word sense) is likely to occur. For example, the verb, *perform* has transitive and intransitive word senses given in the following sentences:

- *The troupe* **performed** *the ballet.*
- *The car* **performs** *well on curves.*

A program could distinguish between these word senses by knowing that the transitive word sense has an associated *object* role while the intransitive word sense does not.

Parsing compounds into constituents is another important problem in interpreting compounds. The ideas given in this report could be incorporated into a program for parsing and interpreting compounds, but further research in this area is necessary.

Currently, the interpretation algorithm returns all possible roles. It could be extended to determine which role is the best when more than one role is returned. Possible extensions would be to incorporate context into the algorithm and to make a hierarchy of the salient roles in the knowledge base.

The interpretation mechanism assumes the relationships between nouns in a compound are roles in the knowledge base. However, these roles do not adequately describe some relationships, as in the compounds *English spelling* and *disk head.* While *disk head* is a lexicalized compound and might be listed in the lexicon, *English spelling* is not a lexicalized compound. If the lexicon listed *English spelling*, then it would have to list *French spelling*, *Spanish spelling*, etc. To interpret compounds such as these, another mechanism is necessary.

Further work is necessary to determine the relationships between words in strictly noun-noun compounds. Looking at examples of these compounds in text could help determine the relationships in which these words are likely to occur.

More studies could be conducted to determine other ways in which the interpretation of nominal compounds could improve precision or recall of IR systems. Specifically, studies could focus on how often synonyms of compounds occur in text, how effective weighting the terms in a query might be, and how

some simple anaphora techniques might be used to recognize the same concept across sentences.

5.2 Summary

This report shows how a knowledge base and an interpretation algorithm could determine the relationships between words in nominal compounds. The approach uses simple techniques and generalizations as much as possible because of the REST knowledge representation and because of the cost/benefit trade-off in information retrieval systems.

The simple algorithm described in this report interprets a majority of nominal compounds using a knowledge intensive approach. The information retrieval experiments indicate the potential for improving the performance of IR systems using the interpretation of nominal compounds. However, the success of the simple technique for recognizing the words in a compound occurring within three words of each other raises questions about the cost-effectiveness of using the interpretation of nominal compounds to improve performance given the need for such detailed knowledge about the words occurring in the compounds. A more cost-effective approach might be to use algorithms to recognize nominal compounds in IR queries and to use simpler techniques (such as recognizing the words in a compound when they occur within three words of each other) to improve the performance of information retrieval systems.

Knowledge representation was one of the most difficult issues in this research. However, the knowledge base designed for this work uses features that any knowledge base must have if it is used for interpreting compounds. The knowledge base is organized around concepts that are organized into categories. Each concept has a set of roles that describe the relationships in which each concept can occur. The roles have semantic preferences to indicate the types of objects that may fill the roles. Salient roles indicate in which relationships a concept is more likely to occur. Finally, nominalizations and role-nominals are identified to facilitate the interpretation of compounds.

This work also shows the importance of nominalizations and role-nominals in both interpreting compounds and in parsing them into constituent phrases. Determining the relationships in strictly noun-noun compounds is harder than in determining relationships in compounds containing a nominalization or role-nominal. The underlying roles in the latter compounds are roles associated with verbs while in the former type of compounds, the relationships are more arbitrary.

References

[1] Yigal Arens, John J. Granacki, and Alice C. Parker. Phrasal analysis of long noun sequences. In *Proceedings of the ACL*, 1987.

[2] Timothy W. Finin. *The Semantic Interpretation of Compound Nominals.* PhD thesis, University of Illinois, Urbana, Illinois, 1980.

[3] D.D. Lewis, W.B. Croft, and N. Bhandaru. Language-oriented information retrieval. *International Journal of Intelligent Systems*, (to appear).

[4] Robert B. Lees. *The Grammar of English Nominalizations.* Indiana University, Bloomington, IN, 1960.

[5] Judith N. Levi. *The Syntax and Semantics of Complex Nominals.* Academic Press, New York, 1979.

[6] J. R. Rhyne. A lexical process model of nominal compounding in English. *American Journal of Computational Linguistics*, 1976. Microfiche 33.

[7] Pamela Downing. On the creation and use of English compound nouns. *Language*, 53:810–842, 1977.

[8] Margaret R. Allen. *Morphological Investigations.* PhD thesis, University of Connecticut, 1978.

[9] David B. McDonald. *Understanding Noun Compounds.* PhD thesis, Carnegie Mellon University, 1982.

[10] Charles Li. *Semantics and the Structure of Compounds in Chinese.* PhD thesis, University of California, Berkeley, 1971.

[11] Pierre Isabelle. Another look at nominal compounds. In *Proceedings of COLING-84*, 1984.

[12] Gerard Salton, Edward A. Fox, and Harry Wu. Extended boolean information retrieval. *Communications of the ACM*, 726(12):1022–1036, 1983.

[13] Paul Jacobs and Lisa Rau. Ace: Associating language with meaning. In *Proceedings of the Sixth European Conference on Artificial Intelligence*, Pisa, Italy, 1984.

[14] Robert Wilensky. KODIAK - a knowledge representation language. In *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, Boulder, Colorado, 1984.

[15] W.B. Croft and R.T. Thompson. I^{3}R: A new approach to the design of document retrieval systems. *Journal of the American Society for Information Science*, 38(6):389–404, 1987.