A Measure of Semantic Relatedness for Resolving Ambiguities in Natural Language Database Requests

by

Julia A. Johnson*
and
Richard S. Rosenberg†

90-6

Technical Report 90-6
January, 1990

*Department of Computer Science, University of Scranton, Scranton, PA 18510
†Department of Computer Science, University of British Columbia, Vancouver, B.C. V6T 1W5

Computer Science Department
University of British Columbia
Vancouver, B.C. V6T 1W5
Canada

A Measure of Semantic Relatedness for Resolving Ambiguities in Natural Language Database Requests

Julia A. Johnson
Department of Computer Science
University of Scranton
Scranton, PA 18510

Richard S. Rosenberg
Department of Computer Science
University of British Columbia
Vancouver, B.C., Canada V6T1W5

January 11, 1990

Abstract

A measure of semantic relatedness based on distance between objects in the database schema has previously been used as a basis for solving a variety of natural language understanding problems including word sense disambiguation, resolution of semantic ambiguities, and attachment of post noun modifiers. The use of min/max values which are usually recorded as part of the process of designing the database schema is proposed as a basis for solving the given problems as they arise in natural language database requests. The min/max values provide a new source of knowledge for resolving ambiguities and a semantics for understanding what knowledge has previously been used by distance measures in database schemas.

Keywords and phrases natural language understanding, natural language interfaces, semantic ambiguity, word sense ambiguity, modifier attachment, database schema, conceptual schema.

1 Introduction

The concern with facilitating communication with databases has been a considerable motivation for the development of natural language interfaces. It is generally agreed that most common database query languages such as SQL are inadequate for a number of reasons including the awkwardness of their syntax and the difficulty (for naive users) of learning their intricacies. Both of these problems would be alleviated if we could communicate with a database using our own natural language (NL).

A great deal of research has been done to automatically generate interpretations for natural language requests that would be considered by humans to be possible in the given domain. Examples include case grammars (Filmore [5]), semantic grammars (Hendrix et al. [11]), and Woods' ATN grammar coupled with a taxonomic lattice [23]. Each uses some source of knowledge about the domain for determining likely interpretations. For natural language interfaces to databases, there has been a great deal of interest [10, 12, 22, 16] in utilizing the database system to provide that knowledge. Previous approaches capture knowledge from the relational database (RDB) schema, but were unsatisfactory for the following reasons: 1.) RDB schemas contain referential ambiguities which seriously limit their usefulness as a knowledge representation strategy for NL understanding. 2.) Knowledge captured from the RDB schema is sensitive to arbitrary decisions made by the designer of the schema. In our work we provide a new solution by applying a conceptual model for database design to the design of a portable natural language interface (NLI). The conceptual model is the

SET model [7, 8]. A heuristic, called a semantic relatedness measure, is introduced which captures knowledge from the SET schema to solve specific linguistic problems all of which involve multiple interpretations of natural language requests. The possible interpretations for a request are ordered by the semantic relatedness measure from most likely to least likely, with some interpretations ruled out completely.

The remainder of this paper is organized as follows: In Section 2 terminology and background concepts are introduced. Section 3 gives an overall design of our proposed system. In Section 4 our heuristic is presented, and in Section 5 it is applied to the problems of prepositional phrase attachment, word sense disambiguation, and semantic ambiguities. A summary of the results of the paper is presented in Section 6.

2 Terminology and Background

2.1 Terminology

Unfortunately, many terms have previously been used in both the areas of natural language and database, but with different meanings. For purposes of clarity we will keep terminology in the two areas separate, and where necessary to avoid confusion, introduce our own terms.

2.1.1 Domain

In the area of relational database systems the term *domain* means a set of values associated with a column of a relation. All values in a column are constrained to be members of its domain.

In the area of natural language understanding the term *domain* is taken to mean the subject that forms the content of the dialog between a person who formulates natural language requests and the system that understands and responds to those requests. The domain of a natural language interface comprises a collection of facts from which answers to database requests are drawn.

To avoid confusion we use the term *domain* in the sense that it is used in the area of natural language understanding, and we use the term *value set* to mean *domain* in the relational database sense.

2.1.2 Natural Language

A sentence constituent that appears in a sentence's parse tree as a terminal node is called a *primitive constituent.* Primitive constituents are not necessarily single English words. For example, the adjective "Computer Science" is a multiple word primitive constituent in the noun phrase "the Computer Science department".

An *internal representation* (IR) for a natural language request is a statement of the conditions under which the NL request is satisfied [1]. A possible IR for the request "Print employees who live in Vancouver" is illustrated in Figure 1. The IR does not specify whether the result should be counted, evaluated as true or false, or printed.

Insert Figure 1 here

2.2 Important Linguistic Problems

In this section three specific linguistic problems that must be handled by any reasonably robust NLI are presented. They are prepositional phrase attachment, semantic ambiguities, and word sense disambiguation. The thesis advanced in this paper holds that the SET schema is a superior source of knowledge over the relational schema for solving the given problems.

2.2.1 Prepositional Phrase Attachment (PPA)

There are a variety of situations in natural language where one sentence constituent modifies another. Consider the noun phrase "the book on the table with a red cover". There are two possible parses for the phrase as illustrated in Figure 2. In the figure nonterminal symbols NP, DET,PP, N, and PREP stand for noun phrase, determiner, prepositional phrase, noun, and preposition, respectively. In P1 the phrases "on the table" and "with a red cover" *both* modify the head noun "book". In P2 the phrase "with a red cover" modifies the noun "table", and the phrase "on the table" modifies the head noun "book".

Insert Figure 2

The sentence constituent being modified is called the *referent*, and the one doing the modification the *modifier*. The PPA problem is to select the most appropriate attachment for the prepositional phrases.

2.2.2 Semantic Ambiguities (SA)

Semantic ambiguities arise when a single syntactic structure (parse tree) maps into more than one internal representation. For example, in a university domain a request such as "Dr. Lee's students" could have any of the following meanings:

1. Students in the same department as Dr. Lee

2. Students enrolled in courses taught by Dr. Lee

3. Students supervised by Dr. Lee

The SA problem is to select the most appropriate meaning for a natural language request.

2.2.3 Word Sense Disambiguation (WSD)

Word sense ambiguity arises when a word in a sentence has more than one meaning. Given the sentence "Students run programs", for example, the noun *program* might mean either a computer program or a recreational program, in which case there would also be two senses for the verb *run* : to *execute* a computer program, and to *administer* a recreational program. The WSD problem is to select exactly one meaning for each word in the sentence.

2.3 Semantic Relatedness

Semantic relatedness between constituents of a natural language request is the closeness in meaning between the constituents, and it is dependent on the domain of discourse. For example, the nouns *dog* and *bone* may be more semantically related than the nouns *dog* and *computer* in a domain that is concerned with the care and feeding of dogs. If, however, the domain is concerned with communication patterns, then the nouns *dog* and *computer* may be more semantically related because humans can communicate with dogs and computers, but not bones.

Semantic relatedness measures have been used for resolving ambiguities in natural language database requests. The denotational part of the meaning of a constituent (as opposed to the procedural part) is typically [9, 15, 4] an object in the database schema. Semantic relatedness between constituents is estimated by measuring the *distance* through the database schema between the objects denoted by the constituents. Favoured interpretations are those which lead to the greatest semantic relatedness between constituents.

2.3.1 Semantic Relatedness Measures in Relational Schemas

Distance between relations in a relational schema is usually measured by counting the number of links required to join the relations together. In this paper, the notion of a link between relations is expressed in terms of more fundamental concepts which permits a better understanding of what is being measured by semantic relatedness measures based on links. A link corresponds to a join condition in the relational query that produces the join. A join condition is a statement $R.A = S.B$ in the query that links relations R and S together by specifying a join of R on column A with S on column B.

In general, the design of a relational schema is arbitary in the sense that two different designers may design different schemas for the same domain. This arbitrariness affects the results of the measure. For example, a common way for two relational schemas to differ is that one relation is used in one schema to express some information while two relations are used in the other schema to express the same information. The join path consisting of the one relation and the join path consisting of the two relations are not of the same length. In our work, a measure of semantic relatedness is proposed which operates on SET schemas [7] and is very insensitive to the arbitrariness of the design of the schema.

2.4 The SET Conceptual Model

The fundamental notions in the SET model are those of set and ordered pair. The concept of set is used in other models such as the Entity-Relationship (ER) model [3]; however, in the ER model the notion of set is an intuitive one. In the SET model the notions of set and ordered pair are based on the provably consistent set theories of [6] and, hence, a sound foundation is provided upon which the richer models of data needed for NL applications can be built.

The *intension* of a set is a property that determines membership in the set. The *extension* of a set is the membership of the set (the collection of objects that satisfy the intension of the set).

The extension of a set changes with time. Given that, the notions of subset and Cartesian product need to be clarified. We understand these notions as they apply to set extensions. If set $S1$ is a subset of set $S2$ then at every time instance t the extension of $S1$ at time t is a subset of the extension of $S2$ at time t. Similarly, at every time instance t the extension of the Cartesian product $(S1 \times S2)$ of sets $S1$ and $S2$ is the Cartesian product of the extensions of $S1$ at time t and $S2$ at time t.

A *binary association* is a subset of the Cartesian product $(L \times R)$ of two sets L and R. L is called the left parent and R the right parent. The extension of an association with left parent L and right parent R is a mapping from the extension of L to the extension of R.

In the following discussion the names A and B refer to set extensions. A mapping in addition to being one-to-one, one-to-many, or many-to-many may be either *in* or *on* the source extension and either *into* or *onto* the target extension. In a mapping from A to B the source extension is A, and the target extension is B. The mapping is *on* A if every member of A occurs in the mapping, and it is *in* A otherwise. The mapping is *onto* B if every member of B occurs in the mapping, and it is *into* B otherwise. *In* and "into" mappings are also referred to as *partial* mappings, and *on* and *onto* mappings as "total" mappings.

The min/max values provide a notation for specifying the 16 different types of associations arising from the mathematical notions of mapping type and total/partial mapping. Associated with each of the source and target extensions of an association is pair of values (p, q) where p has the value either 0 or 1, and q the value either 1 or n. p is called the min value and q the max value. A min value of 0 specifies an *in* or *into* mapping, and a min value of 1 an *on* or *onto* mapping. A max value of 1 specifies a *one* mapping on the source or target extension, and a max value of n a *many* mapping.

The min/max values of an association apply to its extension at every instance in time. That is, at every instance in time the extension of an association is a mapping of the type specified by the association's min/max values. The min/max value associated with the left parent L of association X is called the min/max value of X on L, and similarly for the right parent.

The notion of a binary association is easily extended to that of an m-ary association, $m \geq 2$. An m-ary association has m min/max values, one on each parent set. Usually only the values 0 and 1 are used for the min value and 1 and n for the max value. More general values for min and max are proposed in [17, 19, 21].

2.4.1 The SET Schema

Conceptual modeling using the SET model requires that information *about* the information of interest to the enterprise be recorded by describing a collection of sets. Every set must be explicitly declared which involves giving it a name and supplying other information such as its *intension* and *min/max values*. *Primitive sets* are those whose members are not drawn from some previously declared set. Every non-primitive declared set is a subset of the Cartesian product of one or more previously declared sets. The SET schema comprises all the declared sets of an enterprise.

The intension of a set may be expressed either in a natural language intended to be read by humans or in a formal language that can be interpreted by a machine. Sets with the former type of intension are called *base sets* and with the latter type are called *defined sets*. Value sets such as those usually provided by a programming language (integer, real, character) are assumed to be pre-declared and each is considered to be a defined set. The language DEFINE is introduced in [7] as part of the SET model for expressing the intensions of non-primitive defined sets. The key point to recognize about base sets is that they cannot be defined in terms of other declared sets. The parent sets of a base set may be defined sets, but the selection of its members from the Cartesian product of its parents sets requires human intervention. All non-value primitive sets are base sets.

The advantage of defined sets for portable natural language interfaces is a greater simplicity in the semantic rules used for building the internal representations of requests, and hence, an enhancement in portability. The semantic rules are simplified because the name of a defined set and not its full definition, appears in every rule which uses the set. If the full definition must appear, then the rules are longer and they contain redundant segments.

2.4.2 The Domain Graph

A domain graph (DG) is a graphical representation of the SET schema. The nodes of the graph are labeled with the names of sets and associations. An association X with left parent L and right parent R is denoted by a pair of edges one directed from L to X and the other from R to X. Min/max values for associations are given as labels on the edges of the DG. The min/max value of X on L labels the edge $L \longrightarrow X$, and the min/max of X on R the edge $R \longrightarrow X$. We will refer to an edge in the DG labeled with min/max value (p, q) as a (p, q)-edge.

It is assumed that min/max values are available for labeling the edges of the DG. A second assumption is that the min/max values assigned to the edges are consistent; that is, that extensions exist for the sets named in the DG for which the constraints expressed by the min/max values are satisfied.

3 Overall Design

It has been our observation that the process used for adapting the natural language interface (NLI) to a new subject area and database (DB) overlaps considerably with the process of designing the database schema. Based on this important observation, a design for a portable natural language interface is proposed which features sharing of knowledge about the relationships in the subject of discourse for database schema design and NL understanding.

Different demands are imposed on a knowledge representation strategy by the NLI and the DB system. The NLI needs a language for representing the meaning of NL requests that is independent of the structures of the DB. The DB system, on the other hand, is concerned with structures for representing data efficiently. To solve this dilemma, it is of great importance to choose the appropriate knowledge representation strategy.

Figure 3 gives an overall design strategy of our proposed system. Here the arrows denote information flow. It is particularly noteworthy that the SET schema not only provides knowledge for constructing the relational DB schema (previously researched by Gilmore [8] and Storey and Goldstein [18]) but more significantly provides knowledge for the purpose of adapting the NLI to a new domain.

Insert Figure 3 here

By portability of an NLI we mean the extent to which it can be adapted to a new domain and database, by the database administrator (DBA), as opposed to a linguistic expert. Our design strategy results in an enhancement of portability. It is the responsibility of the DBA to produce the SET schema and to design the relational schema based on information in the SET schema. The DBA is also charged with the task of adapting the NLI to the given domain and database. Little further work is required for the DBA to provide the NLI with knowledge needed for natural language processing, because the knowledge has already been gathered for the purpose of designing the RDB schema.

4 A Measure of Semantic Relatedness in SET Schemas

Our semantic relatedness measure in SET schemas is an elaboration of a measure in Entity-Relationship (ER) [3] schemas previously proposed by Wald and Sorenson [20]. In the ER model, an association may be one-to-one, one-to-many, or many-to-many. Here we extend Wald and Sorenson's measure to include information about whether an association is *in* or *on* and *into* or *onto*.

The remainder of this section is organized as follows: In subsection 4.1 our semantic relatedness measure is briefly described and the main motivation for its design, that it be unaffected by arbitrary decisions in the design of the SET schema, is discussed. In subsection 4.2 an sample domain is described to serve as a source of examples throughout the remainder of the paper. In subsections 4.3 and 4.4 the relationship between the min/max values and word meanings is examined. Our semantic relatedness measure based on min/max values is presented in subsection 4.5.

4.1 Sensitivity of the Heuristic to SET Schema Design Alternatives

A desirable feature of any heuristic that operates on database schemas is that it should be invariant with respect to arbitrary decisions made by the designer of the schema. Semantic relatedness measures in relational schemas are sensitive to the arbitrariness of the design of the relational schema. The design of SET schemas is arbitrary as well, but our measure is very insensitive to it.

In our work, each interpretation of a natural language request is represented as a subgraph of the domain graph. Edge weights are derived from the min/max values that label the edges, and interpretations are compared by comparing the weights of their subgraphs.

The choice of weights is driven by the requirement that the heuristic should give the same outcome independent of arbitrary decisions in the design of the SET schema. It was found that a necessary condition for this requirement is that (1,1)-edges have a weight of zero. To illustrate, consider the marriage association which may be represented either as an entity or as a relationship. Schemas for the two alternatives are illustrated in Figure 4.

Insert Figure 4 here

In schema (a), a marriage is considered to be a relationship. Association *Married* is defined as a subset of the Cartesian product ($Male \times Female$) of two sets *Male* and *Female*. A male may not be married, but if he is, he is married to at most one female, and the same holds for females. These constraints are expressed by the min/max values of *Married* on *Male* and on *Female*.

In schema (b), a marriage is considered to be an entity. If male m and female f are married and that marriage is represented by the entity mf, then the pair (m, mf) is a member of *Husband* and the pair (mf, f) is a member of *Wife*. Since every marriage has both a husband and a wife, the min values of *Married* on *Husband* and on *Wife* are both 1. Since every marriage has at most one husband and at most one wife, the max values of *Marriage* on *Husband* and on *Wife* are both 1.

Although they look different, the two schemas express the same information. Likewise, the domain graphs (DGs) for the two schemas have the same weight. The two new edges that are introduced in Schema (b) are both (1,1)-edges each of which has a weight of 0. Otherwise, the edges in the two DGs are identical and their weights are identical because the two new edges contribute nothing to the weight.

In [14], the subject of the sensitivity of our heuristic to arbitrary decisions in the design of the SET schema is examined in detail. The choice of weights for edges other than (1,1)-edges is arbitrary in our heuristic, but we have found that there is, in fact, a wide range of weights that could have been assigned without affecting the outcome of the heuristic. This subject is addressed in Section 5.3 of this paper.

4.2 An Example from the University Domain

To facilitate our presentation of a heuristic for measuring semantic relatedness, a domain graph for the University domain is considered (Figure 5). The entities of interest are students, courses, professors, and departments. The associations of interest are as follows:

SC associates with a student the courses that he or she is taking

CP associates with a course the professor who teaches the course

SD associates with a student the department in which he or she is registered

PD associates with a professor the department in which he or she works

Sup associates with a student the professor who supervises the student's research

CN associates with a course the name of the course

CPSC_Course is the set of courses offered by the computer science department. The sets $CName$, $Grade$, $SName$, and $PName$ are value sets which contain course names, grades, student names, and professor names, respectively.

Insert Figure 5 here

4.3 What is a Word Meaning?

The primitive constituents of a request map to vertices in the DG. The mapping is specified as part of the process of adapting the natural language interface to a new domain, and it gives the meanings (denotations) of the primitive constituents. Some primitive constituents do not denote vertices in the DG. Examples include noise words ("please" and "quickly", as in "Please print the good students quickly") which can be ignored without changing the meaning of the request and determiners ("a", "the", "some", "all") which map into restrictions on vertices. Some primitive constituents denote more than one vertex, and this is the source of word sense ambiguities in natural language database requests.

In the examples given here, the following rules govern the assignment of meanings to primitive constituents.

1. Database values ("CPSC101", "Dr. Lee") denote value sets (*CName*, *PName*, respectively)

2. Nouns (student, course, department) denote non-value sets (*Student*, *Course*, *Dept*).

3. Verbs (take, teach, receive) denote associations (*SC*, *CP*, *SCG*).

4.4 The Relationship between Min/Max Values and Word Meanings

In this discussion, we will focus on disambiguating prepositional phrase (PP) attachments that use the prepositions "with" and "in" and on choosing the most appropriate meaning for pre-noun modifiers (in particular, possessives).

The preposition "with" has many different meanings. The heuristic developed here deals with only one of them - the *part of* relationship which involves an "inseparable possession", or "possession by nature, not accident" [13]. Examples include "fish with bones", "vase with handles", "man with sinister expression", and "holiday with all expenses paid". Note that fish bones do not exist without the fish and the bones belong to no fish other than the given one, the handles belong to the vase and no other, the sinister expression exists as part of the man and the same expression will not exist as part of any other man, and the paid expenses are not relevant except with respect to the holiday.

The min/max values represent the *part of* relationship by what has become known as an existence dependency association. Given an association X between A and B, the set B is said to be *existence dependent* on A if an entity in B cannot exist independently from an entity in A. (e.g., a volume of a book is existence dependent on the book, a ward of a hospital is existence dependent on the hospital). If X with parent sets A and B is an existence dependent association, then the min/max values of X on the dependent set B are (1,1). Note that not all associations with min/max values (1,1) are existence dependent associations.

The preposition in "in" also has a *part of* meaning. Examples include the phrases "in bad health" and "in danger". Bad health exists only with respect to the person who is suffering from it, and danger does not exist independent of the object of the danger (e.g. a species, a country).

A weaker form of relationship is the *exclusive association.* Given an association X between A and B, the set B is exclusively associated with A if the min/max value of X on B is (0,1).

Some pre-noun modifiers that indicate possession (e.g., Dr. Lee's students) denote an exclusive association between the sets denoted by the noun and the modifier. (The set denoted by the noun is exclusively associated with the set denoted by the modifier.) If the interpretation for the phrase is "Students whose research supervisor is Dr. Lee, and assuming that in the given domain each student has at most one research supervisor, then the phrase denotes an exclusive association.

Weaker forms of pre-noun modifiers (e.g., Jones' courses) are possible. In the interpretation "courses taken by Jones", although Jones takes the courses, they may also be taken by others. Such modifiers denote associations that are neither exclusive nor existence dependent.

An assumption underlying our heuristic is that when a database request has more than one possible interpretation, those that denote existence dependencies are more likely than those that denote exclusive associations, which are in turn more likely than those that denote neither existence dependent nor exclusive associations.

4.5 Query Graphs

To provide a measure of relatedness between primitive constituents of a natural language request we use the notion of a *query graph* which has previously been used as a means of representing database queries [20, 2]. Here the query graph is used as a means of representing natural language requests. Since NL requests are far more complex than DB queries, it is necessary for us to restrict the complexity of the NL requests under consideration.

A natural language request is *simple* if it requests information about a collection of related entities. Examples of simple requests in the university domain are:

1. a professor in a department with a student who takes a course named CPSC101

2. a student who received a grade of 'B' in CPSC101

In request (1), for example, a relationship exists between professors and departments, departments and students, students and courses, and courses and course names. An example of a request that is not simple is the conjunction of the above two requests: a professor in a department with a student who takes a course named CPSC101 and a student who received a grade of 'B' in CPSC101. The student referred to in the left piece of the request bears no relationship to the student referred to in the right piece.

Following the terminology of [20] we use the terms "target graph" and "query graph" but extend their meanings to apply to natural language database requests rather than database requests expressed in a formal query language. The *target graph* for a simple request Q, $TG(Q)$, is the set of vertices denoted by primitive constituents of Q. A *query graph* for Q is any subgraph of the DG that

1. is a tree each leaf node of which is contained in $TG(Q)$

2. contains the vertices in $TG(Q)$.

Not every simple request can be represented by a tree. For example, the following request would be represented by an undirected-cyclic subgraph of the university domain graph (Figure 5).

a student in a course taught by a professor
who is the student's research supervisor

Such requests are referred to as *cyclic requests*. A cyclic request is represented as a collection of trees by removing for each cycle one of the edges that creates the cycle. Since there is for each cycle more than one edge whose removal will break the cycle, there will be more than one possible resulting tree. A cyclic request is represented by the collection of all such possible trees (a forest).

Each edge of a query graph is labeled with a min/max value from which a weight for the edge and, therefore, a weight for the query graph can be computed. The weight of an edge labeled with min/max value (p, q) is calculated as follows:

1. If $p = q = 1$, then the weight is 0.

2. If $p = 1$ or $q = 1$, then the weight is 1.

3. If $p = 0$ and $q = 0$, then the weight is some large value such as the number of vertices in the query graph.

Given a tree with root v and directed edges, the *forward edges* relative to v are the edges that point away from v.

The weight of a query graph G relative to $v \in TG(Q)$ is the sum of the weights on forward edges relative to v. The absolute weight (or simply the weight) of G is the minimum of the relative weights over all $v \in TG(Q)$.

Example 1. Given the University DG, an example of a query graph for $TG(Q) = \{Student, Prof\}$ is $(Student \longrightarrow SC \longleftarrow Course \longrightarrow CP \longleftarrow Prof)$. The forward edges relative to $Student$ are $(Student, SC)$ and $(Course, CP)$. The weight relative to $Student$ is the sum of the weights on the two forward edges relative to $Student$ $(5 + 1) = 6$. (The number of vertices in the query graph is 5.) The forward edges relative to $Prof$ are $(CP, Prof)$ and $(SC, Course)$. The weight relative to $Prof$ is $(5 + 5) = 10$. The absolute weight is the minimum of the weights relative to $Student$ and relative to $Prof$. Therefore, the weight of the query graph is 6. The weight of a cyclic query graph is the weight of the minimum weight tree among those in the forest trees that represent it.

Possible interpretations for a request are ordered based on the weights of their associated query graphs. The lower the weight, the more likely the interpretation is considered to be. A query graph with many $(0, n)$-edges will have a large weight. Our heuristic requires knowledge to be useful. If all of the query graphs for a request have many $(0, n)$-edges, then the measure provides little basis for comparison.

5 Use of Min/Max Values for Resolving Ambiguities

In this section the use of the min/max values for resolving ambiguities in natural language database requests is demonstrated by providing examples from two different domains. The first is the University domain which has been described in Section 4, and the second is a medical domain [19]. The section concludes with an analysis of the extent to which the parameters of the heuristic can be varied without affecting its outcome.

5.1 The University Domain

The examples presented in this subsection refer to the domain graph of Figure 5. The three types of ambiguity are semantic, word sense, and prepositional phrase attachment.

5.1.1 Semantic Ambiguity

Let $TG(Q)$ be the set of vertices referenced by simple request Q. The semantic ambiguity (SA) problem is to select from among the query graphs determined by $TG(Q)$ the one that corresponds with the best interpretation for Q. The approach presented here, like that of Wald and Sorenson but with a different weight measure, is to select the interpretation that corresponds with the query graph of smallest weight where the weight of a query graph is the minimum of the relative weights over all $v \in TG(Q)$.

Example 2. Among the possible interpretations, for the phrase "Dr. Lee's student's" only the most likely two are being treated here:

1. "Students taught by Dr. Lee"

2. "Students supervised by Dr. Lee"

Internal representations for the two interpretations espressed in the language DEFINE follow:

1. [**For some** x:*Student*] [**For some** y:*Course*][**For some** z:*Prof*]

   $(< x, y >:SC$ **and** $< y, z >:CP$ **and** $< z,$"Dr.Lee"$>:PN)$

2. [**For some** x:*Student*] [**For some** y:*Prof*]

   $(< x, y >:Sup$ **and** $< y,$"Dr.Lee"$>:PN)$

Insert Figure 6 here

$TG(Q) = \{Student, PName\}$ determines two query graphs. The path between *Student* and *PName* through *Course* and *Prof* corresponds with interpretation (1), and the path between *Student* and *PName* through *Sup* corresponds with interpretation (2). Figure 6 illustrates the calculations for choosing the best interpretation. Recall that $(0, n)$-edges have a weight equal to the number of nodes in the query graph, and $(0, 1)$ and $(1, n)$-edges have a weight of 1. Only forward edges relative to vertex v are counted when computing the weight relative to v. For computing the weight of the query graph for interpretation (1) relative to *Student*, for example, there are three forward edges relative to *Student* with weights 7 (the number of nodes in the QG), 1, and 0 in left to right order. In future examples, detailed calculations of the relative and absolute weights will not be given, since the calculations are straight-forward.

5.1.2 Word Sense Disambiguation

Each primitive constituent of a request denotes 0, 1, or more vertices in the DG. For request Q, a target graph is obtained by selecting exactly one vertex from each nonempty set of vertices denoted by primitive constituents of Q. Word sense disambiguation is the problem of selecting the best target graph $TG(Q)$.

Example 3. Consider the request "Jones' courses" and suppose that "Jones" could be the name of either a student or a professor. Furthermore, assume that both *Student* and *Prof* are in the current focus, which is to say that both the student

"Jones" and the professor "Jones" have been recently referred to in the dialog and are therefore possible meanings for the proper noun "Jones". In the university domain if "Jones" names a student, then the request asks for the courses in which Jones is enrolled. Otherwise, it asks for the courses taught by professor Jones. Figure 7 illustrates application of the measure to choose the best meaning for the ambiguous word "Jones". Our heuristic selects *Prof* as the best meaning for "Jones" and, hence, the favored interpretation is "courses taught by professor Jones".

Insert Figure 7 here

5.1.3 Modifier Attachment

We are only beginning to explore the problem of prepositional phrase attachment. In this paper, requests are limited to those which have a maximum of two prepositional phrases. Consideration of requests in which the number of modifiers is greater than two, which would give more possibility for ambiguity, is left for future work.

Example 4. Consider the request "a professor for a course with no students". An internal representation (IR) for the interpretation in which "with no students" modifies "professor" follows:

1. [**For some** x:*Prof*] [**For some** y:*Course*]

$$(< y, x >:CP \textbf{ and}$$

$$[\textbf{For all } z{:}Student] \textbf{ not} < z, x >:Sup)$$

An IR for the interpretation in which "with no students" modifies "course" follows:

2. **[For some** x:*Prof*] **[For some** y:*Course*]

$(<y,x>:CP$ **and**

[For all z:*Student*] **not**$<z,y>$: $SC)$

The nouns "student", "course", and "professor" denote, respectively, the vertices *Student*, *Course*, and *Prof*. To determine the best attachments for the modifiers, we again look at the query graph. However, the target graph for the given request will be the same regardless of modifier attachments and therefore will determine the same query graphs for the two interpretations. It is necessary to distinguish the different attachments of the modifiers. We do this by adding vertices to the target graph (TG) to denote modifier attachments.

The TG for the request is $\{Prof, Student, Course\}$. The TG augmented with vertices to denote the attachment of the phrase "with no students" to the noun "professor" is

$$TG_1 = \{Prof, Student, Course, Sup\}.$$

The TG augmented with vertices to denote the attachment of the phrase "with no students" to the noun "course" is

$$TG_2 = \{Prof, Student, Course, SC\}.$$

Insert Figure 8 here

A query graph for each target graph is given in Figure 8. The query graph for TG_1 has relative weights 6 on *Student*, 6 on *Course*, 10 on *Prof*, and 5 on *SC*. The query graph for TG_2 has relative weights 6 on *Student*, 6 on *Course*, 10 on *Prof*, and 1 on *SC*. The heuristic favors the attachment of the phrase "with no students" to the noun "course".

It is not always possible to denote the attachment of a modifier to a referent by a vertex that already exists in the DG. In general, a path between the vertices denoted by the referent and the modifier must be added to the TG. The handling of the more complex case is illustrated in Example 6 to follow.

5.2 The Medical Domain

Our second source of examples is a medical domain that has been described by Tsichritzis and Lochovsky in [19] and that is a scaled down version of a real application. The entities of interest are hospitals, wards of hospitals, hospital staff, doctors, patients, labs, tests, and diagnoses. The associations of interest are described in Figure 9 which also gives the domain graph for the medical domain. Tsichritzis and Lochovsky give an ER diagram for the medical domain and the given domain graph is an expansion of that one to include min values for the associations.

Insert Figure 9 here

5.2.1 Semantic Ambiguity

Example 5. Resolution of semantic ambiguity in the medical domain is illustrated in Figure 10. The request "a patient in a hospital" has three interpretations in the medical domain:

1. a patient who occupies a bed in a ward in a hospital

2. a patient who has a doctor on staff at a hospital

3. a patient who has an order for a test at a lab that does work for the hospital

Interpretation (1) is favored over the other two, and (2) is favored over (3).

Insert Figure 10 here

5.2.2 Modifier Attachment

Example 6. Consider the request "a patient in a hospital with tests". The problem here is to determine whether "with tests" modifies "hospital" or "patient". In either case the phrase "in a hospital" modifies the head noun "patient". There is semantic ambiguity in the association between patients and hospitals. For the purpose of the example, we will assume that the ambiguity is resolved in favor of the interpretation "a patient who occupies a bed in a ward of the hospital" based on the results of Example 5. If the phrase "with tests" modifies "patient", then the request is for a patient who occupies a ward of a hospital and for whom medical tests have been ordered. Otherwise, the request is for a patient who occupies a ward of a hospital which has outstanding orders for tests at some lab. Resolution of the ambiguity is illustrated in Figure 11.

Insert Figure 11 here

The target graph for the interpretation in which "with tests" modifies "patient" is $TG_1 = \{Patient, Hospital, Test, PT\}$. The vertex PT has been added to the target graph to denote the attachment of the modifier.

The target graph for the interpretation in which "with tests" modifies "hospital" is $TG_2 = \{Patient, Hospital, Test, Hosp\_Test\}$. The vertex $Hosp\_Test$ denotes the attachment of the modifier which is represented by a set defined as follows:

```
Hosp_Test def
    select  h:Hospital, t:Test
    where   [For some l:Lab](< h, l >:HL
            and < l, t >:LT)
```

$Hosp\_Test$ is the set composition of HL and LT. Min/max values for $Hosp\_Test$ are $(0, n)$ on $Hospital$ and $(0, n)$ on $Test$. These values are computed by taking the product of min/max values on forward edges relative to $Hospital$ (to get min/max on $Hospital$) and similarly to get min/max on $Test$.

A vertex labeled $Hosp\_Test$ is introduced to the DG which introduces a cycle. To avoid cyclic query graphs we break the cycle thus generating a collection of minimal connected acyclic subgraphs each of which contains the vertices in the target graph. There are two such graphs for TG_2 which are illustrated in part 2 of Figure 11. The query graph determined by a target graph is the minimum weight subgraph among the ones that are obtained by breaking cycles in this way.

5.2.3 Analysis

We have seen a variety of examples from different domains that illustrate the use of min/max values for resolving ambiguities in natural language requests. In this section the results of the examples are analysed to gain an understanding of why the heuristic works. Here the term *association* will have its mathematical meaning given in Section 2.4, and the term *relationship* will be used informally to refer to connections between words in sentences or objects in the domain.

The examples have focused on disambiguating prepositional phrase attachments (PPA) that use the prepositions "with" and "in", on choosing the most appropriate meaning for pre-noun modifiers (in particular, possessives), which are semantically ambiguous (SA), and on word sense disambiguation (WSD). Our heuristic is intended to be applied simultaneously to the problems. The different components of the heuristic interact with each other producing better results than if each individual component were to be applied separately. In the words of Jensen and Binot [13] "The cumulative effect of many heuristics, and not the perfection of each one taken separately, has to do the job".

For the purpose of analysis, we can separate out the component of the heuristic that is being used for disambiguating prepositional phrase attachments. The possible attachments are ordered by the extent to which they are *part of* the request as a whole. This is to say that the natural language request is conceived as a whole and the attachment of a modifier as a part of the whole. A measure of the extent to which the attachment of the modifier is *part of* the whole derives from the *part of* relationships that build the whole.

Examples 4 and 6 address the problem of prepositional phrase attachment. For the request "a professor in a course with no students" (Example 4) we find that the attachment of "with no students" to "course" is more *part of* the request than the alternate attachment. In the university domain a student is existence dependent on the professor who supervises his or her research, and one might think that the attachment of "with no students" to "professor" would be indicated. However, our heuristic is concerned with the extent to which the PP is *part of* the request as a whole, and this occurs when the PP modifies "course". In Example 6 the request is "a patient in a hospital with tests", and the heuristic indicates that the best attachment of the PP "with tests" is to the noun "patient". In the medical domain a test is existence dependent on the patient being tested, but this is not the reason why the attachment of "with tests" to "patients" is favored. Rather it is that the set denoted by that attachment is exclusively associated with the set that denotes the meaning of the request as a whole, whereas a weaker form of relationship exists between the referent and the modifier for the alternate attachment.

The present results cannot be understood independently from other heuristics such as those which track topic, focus and context and that are necessary for choosing the best interpretation for a request. However, an examination of the outcome of the examples is useful for the given requests which exclude many of the linguistic problems for which other heuristics would be needed. A complicating factor is that, since the heuristic relies on global knowledge about the request, it is more useful for lengthy requests than the simple fragments of requests that are illustrated in the examples.

The heuristic captures two different measures of semantic ambiguity. One is the number of real world objects that are implicitly referenced in the request. Implicitly referenced vertices are those in the query graph that are not denoted by primitive constituents. The larger the number of implicitly referenced vertices, the less likely that interpretation is considered to be. Here we are assuming that the most likely interpretation is the least ambiguous one.

In the absence of information provided by min/max values (if all of the edges are labeled (O, n)), the heuristic counts edges. The likelihood of an interpretation decreases as the number of edges (and, since the query graph is a tree, also vertices) increases. For the request "Dr. Lee's students", there are five implicitly referenced vertices (SC, Course, CP, Prof, PN) for the interpretation "Students in Dr. Lee's course", and only three (Sup, Prof, PN) for "students supervised by Dr. Lee", which accounts in part for the latter interpretation being favored.

It is possible for semantic ambiguity to exist when the alternate interpretations have the same number of implicitly referenced vertices. For example, the interpretations "students in Dr. Lee's courses" and "students in Dr. Lee's department" both implicitly reference five vertices. The min/max values provide additional information for distinguishing between such interpretations. The second measure of semantic ambiguity is the extent to which the primitive constituents are related to the request as a whole.

Example 5 features the request "a patient in a hospital" which has three possible interpretations in the medical domain. The interpretations "a patient who occupies a bed of a ward of a hospital" and "a patient who has a doctor on staff at the hospital" implicitly reference the same number of vertices. The former interpretation is favored because the constituent "hospital" is more part of the remainder of the request that it is for the alternate interpretation.

Example 3 illustrates use of the heuristic for resolving WSA. The request is "Jones' Courses" and "Jones" could refer to either a professor or a student. The relationships between alternate meanings for the ambiguous component (a word in this case) and the request as a whole are considered. The strength of the relationsh between an ambiguous word and the request as a whole is measured by the weight of the query graph for a given interpretation relative to the vertex denoted by the ambiguous word. By this measure the interpretations "courses taken by the student Jones' " and "courses taught by professor Jones" lead to relationships between the component "Jones" and the request as a whole that are of equal strength. Thus, the two interpretations are considered to be equally likely. The results of section 5.1.2, however, report a preference for the interpretation "courses taught by professor Jones".

The discrepancy arises because the above measure separates out only that component of the heuristc that is concerned with word sense ambiguity. Our strategy of using the minimum of the relative weights over all vertices in the target graph is intended as a way of collapsing the three measures of ambiguity into one heuristic. If only WSA is being considered, then the weight relative to one particular vertex, specifically the one denoted by the ambiguous word, is of interest. The request "Jones's courses" is a case where the heuristic overgeneralizes, and thus incorrectly favors the interpretation "courses taught by professor Jones".

In conclusion, the min/max values model the *part of* construction in the English language ("petals of a flower", "handles of a vase", "wards of a hospital", "copies of a book") as well as weaker forms of possessive relationships. In fact, they provide a metric for measuring the strength of the possession (e.g.,"petals of a flower" is stronger than "a professor's students"). Our heuristic reformulates the *part of* relationships that are referenced in a natural language request as a *part of* relationship between the request itself and the ambiguous components of the request. The favored interpretation is the one in which there is the strongest *part of* relationship between the ambiguous components and the request itself.

5.3 Varying the Parameters of the Heuristic

In this subsection, the effects of varying the parameters of the heuristic are investigated. The parameters are the weights assigned to the different types of edges. The analysis presented briefly in Section 4.1 and in greater detail in [14] provides convincing evidence that the weight of a (1,1)-edge should be 0. For the other types of edges, let their weights be denoted by variables X, Y, and Z as illustrated in the following table:

min/max	weight
$(1,1)$	0
$(0,1)$	Y
$(1,n)$	Z
$(0,n)$	X

Insert Figure 12 here

Figure 12 illustrates the computations of semantic relatedness for the phrase "Dr. Lee's Students" using variables X, Y, and Z (assumed to be non-negative integers) for the edge weights.

The computations using specific values for the edge weights have been illustrated in Figure 6. To obtain the same relative ordering of the query graphs that was obtained using the specific values the following conditions must hold:

1. $\mathrm{X} + \mathrm{Z} \leq 3\mathrm{X}$

2. $\mathrm{X} + \mathrm{Z} > 0$

3. $2\mathrm{X} \geq 0$

Thus, if $\mathrm{X} \geq 1$, the same relative ordering of the interpretations is obtained for the given request.

In the appendix, a similar set of inequalities is given for each of the requests that has been studied in Section 5. In addition, the constraints on the edge weights that can be derived from each set of inequalities are stated. The results indicate that for the sample requests the same relative ordering on interpretations would have been obtained if the conditions $(\mathrm{X} > 0)$ and $(\mathrm{Z} > 0)$ hold.

6 Conclusion

In this paper we have focused on linguistic problems concerned with ambiguities in natural language database requests, and we have proposed a design strategy for natural language interfaces which involves capturing knowledge for resolving ambiguities from the SET schema.

The main contributions of this work to ongoing research are as follows:

1. The discovery that the process used for adapting the natural language interface to a new domain and database overlaps considerably with the process of designing the DB schema.

2. The development of heuristics based on the knowledge referred to in (1) for resolving ambiguities in natural language database requests, in particular, semantic ambiguity, word sense ambiguity, and prepositional phrase attachment.

3. Specification of the relationship between the mathematical notions of mapping type and total/partial mapping and specific English language constructions.

Further research is expected, particularly with regard to 3. The richness of description that the notions of mapping type and total/partial mapping provide for modeling language phenomena has not been fully tapped by our approach. We have provided a general heuristic that applies to several linguistic problems and several English language constructions. A challenging problem for the near future is to provide heuristics based on the notions of mapping type and total/partial mapping, but specialized to each linguistic problem and language construction.

Appendix. Varying the Parameters of the Heuristic

This appendix provides material to support the conclusion that the parameters of our heuristic can be varied considerably without affecting its outcome. Inequalities expressing the computation of semantic relatedness for the possible interpretations for each of several requests is given. The edge weights are left as variables X, Y and Z in the computations. Our objective is to determine whether the particular assignment of edge weights used in the heuristic is necessary and, if not, the extent to which the given edge weights can be varied without affecting the results. The requests, their associated sets of inequalities, and the conclusions about edge weight assignments that can be derived from the inequalities follow:

Jones' courses

1. $\mathrm{X} \leq 2\mathrm{X}$

2. $\mathrm{Z} \leq 2\mathrm{X}$

3. $\mathrm{Z} < \mathrm{X}$

Conclusion: $\mathrm{X} > 0, \mathrm{X} > \mathrm{Z}$.

A professor for a course with no students

1. $\mathrm{X} \leq 2\mathrm{X}$

2. $\mathrm{X} \leq \mathrm{X} + \mathrm{Z}$

3. $\mathrm{Z} \leq 2\mathrm{X}$

4. $Z \leq X + Z$

5. $Z < X$

Conclusion: $X > 0$.

A patient in a hospital

1. $Y \leq X + Z$

2. $X + Z \leq 2X$

3. $(Y < X + Z) \vee (Y < X)$

4. $X < X + Z$

Conclusion: $Z > 0$, $X > 0$.

A patient in a hospital with tests

1. $Y \leq X + Y$

2. $Y \leq 2X + Z$

3. $2X + 2Y \leq 3X + 2Z$

4. $2X + 2Y \leq 3X + Z$

5. $2X + 2Y \leq 2X + Z$

6. $3X + 2Z \geq 2X + Z + Y$

7. $3X + Z \geq 2X + Z + Y$

8. $2X + 2Z \geq 2X + Z + Y$

9. $(2X + 2Y > Y) \vee (2X + Z + Y > Y)$

Conclusion: $X > 0$; if $Y > 0$ then $Z > 0$.

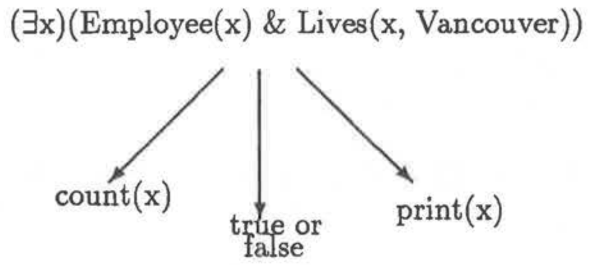


Figure 1: A Possible IR for the Request "Print employees who live in Vancouver"

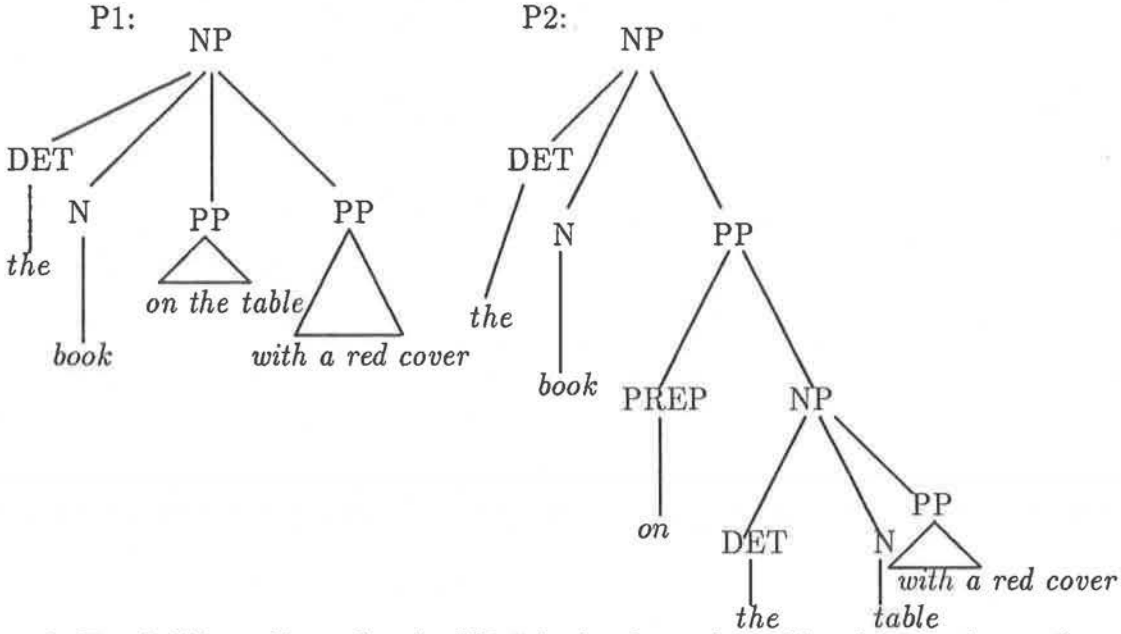


Figure 2: Partial Parse Trees for the NP "the book on the table with a red cover"

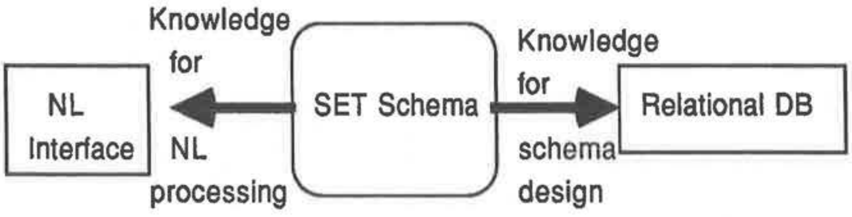


Figure 3: A Data Management Strategy for Portable NLI

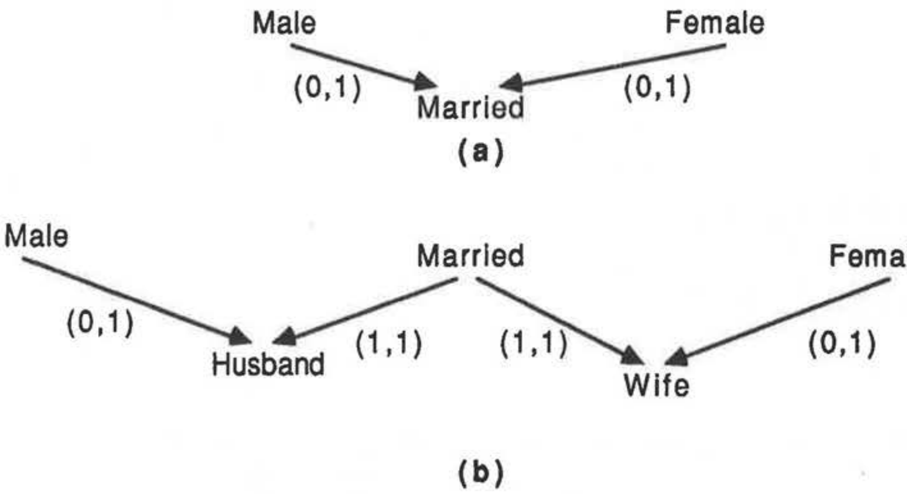


Figure 4: Marriage as an Entity and a Relationship

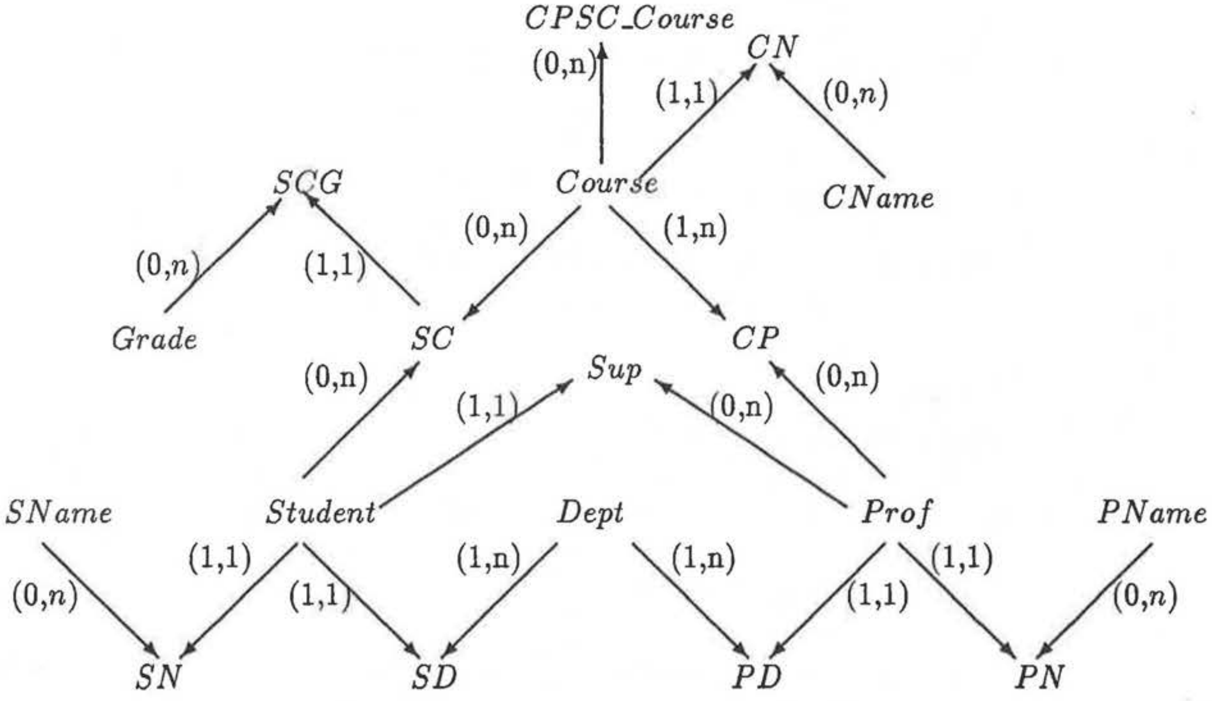


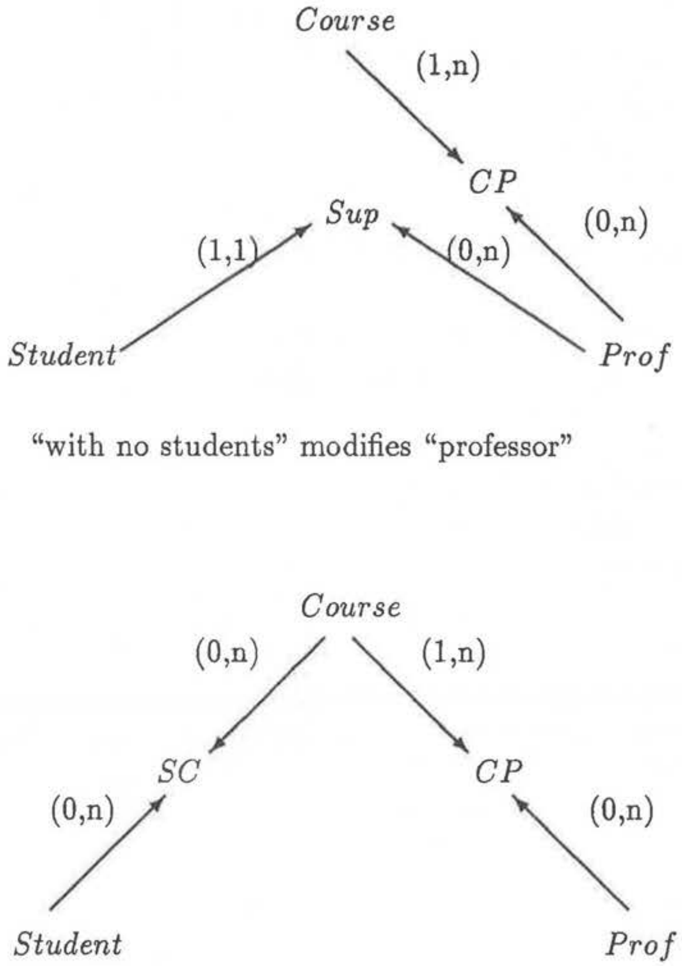
Figure 5: A Domain Graph for the University Domain

```
request: Dr. Lee's students
word  assignments
   "Dr. Lee"  PName
   "student"   Student

  1. Students in courses taught by Dr. Lee

Student--->SC<---Course--->CP<---Prof--->PN<---PName
          (0,n)   (0,n)       (1,n)   (0,n)      (1,1)     (0,n)

   7 nodes
   weight rel. Student   (7+1+0)=8
   weight rel PName       (7+7+7)=21
   absolute weight          minimum(8,21)=8


   2. Students supervised by Dr. Lee

      Student  --->Sup<---Prof--->PN<---Pname
                 (1,1)      (0,n)     (1,1)    (0,n)

    5 nodes
    weight rel Student    (0+0)=0
    weight rel PName       (5+5)=10
    absolute weight          minimum(0,10)=0

    *** favored interpretation is 2 ***
```

Figure 6: Application of Semantic Relatedness Measure to "Dr. Lee's students"

```
request: Jones courses
word  assignments
  "Jones"  SName
  "course"   Course

 1. Courses taken by the student Jones

 SName--->SN<---Student--->SC<---Course
     (0,n)     (1,1)              (0,n)      (0,n)

  5 nodes
  weight rel. Course      5
  weight rel SName      10
  absolute weight          5


  2. Courses taught by professor Jones

     Course   --->CP<---Prof--->PN<---Pnam e
            (1,n)      (0,n)      (1,1)   (0,n)

   5 nodes
   weight rel Course      1
   weight rel PName     10
   absolute weight          1

   *** favored interpretation is 2 ***
```

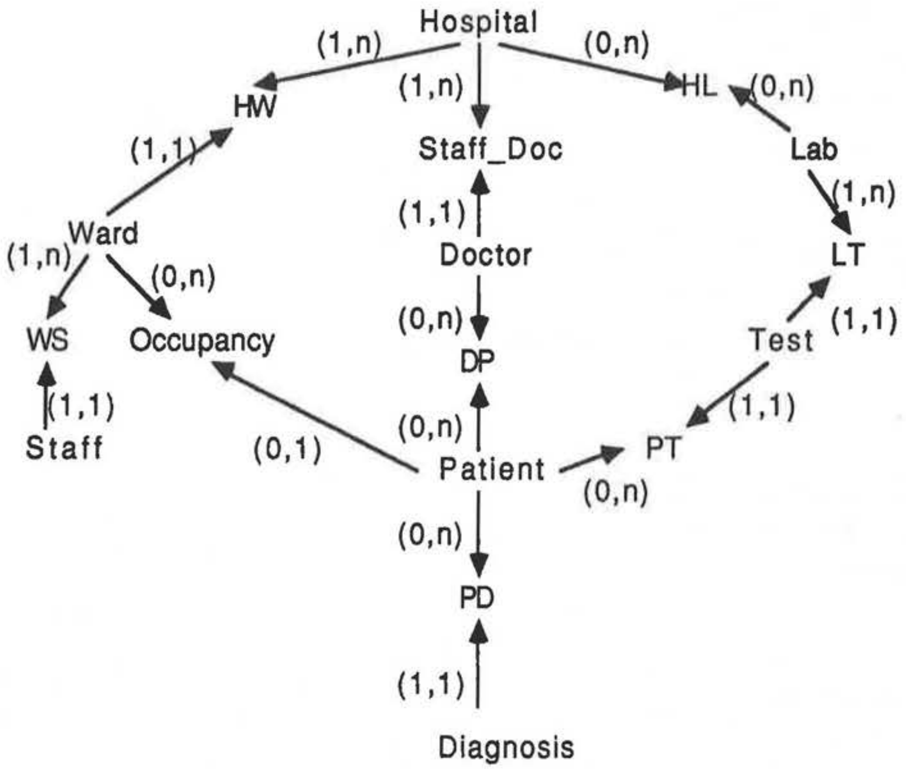
Figure 7: Application of Semantic Relatedness Measure to "Jones' Courses"



"with no students" modifies "professor"

"with no students" modifies "course"

Figure 8: Query Graphs for "a professor for a course with no students"



HW - associates the wards of a hospital with the hospital

WS - associates employees who work in a ward with the ward

Occupancy - associates with a ward the patients that occupy a bed in the ward

PD - associates with a patient the medical diagnosis(ses) reached for the patient

DP - associates with a patient his or her attending doctor(s)

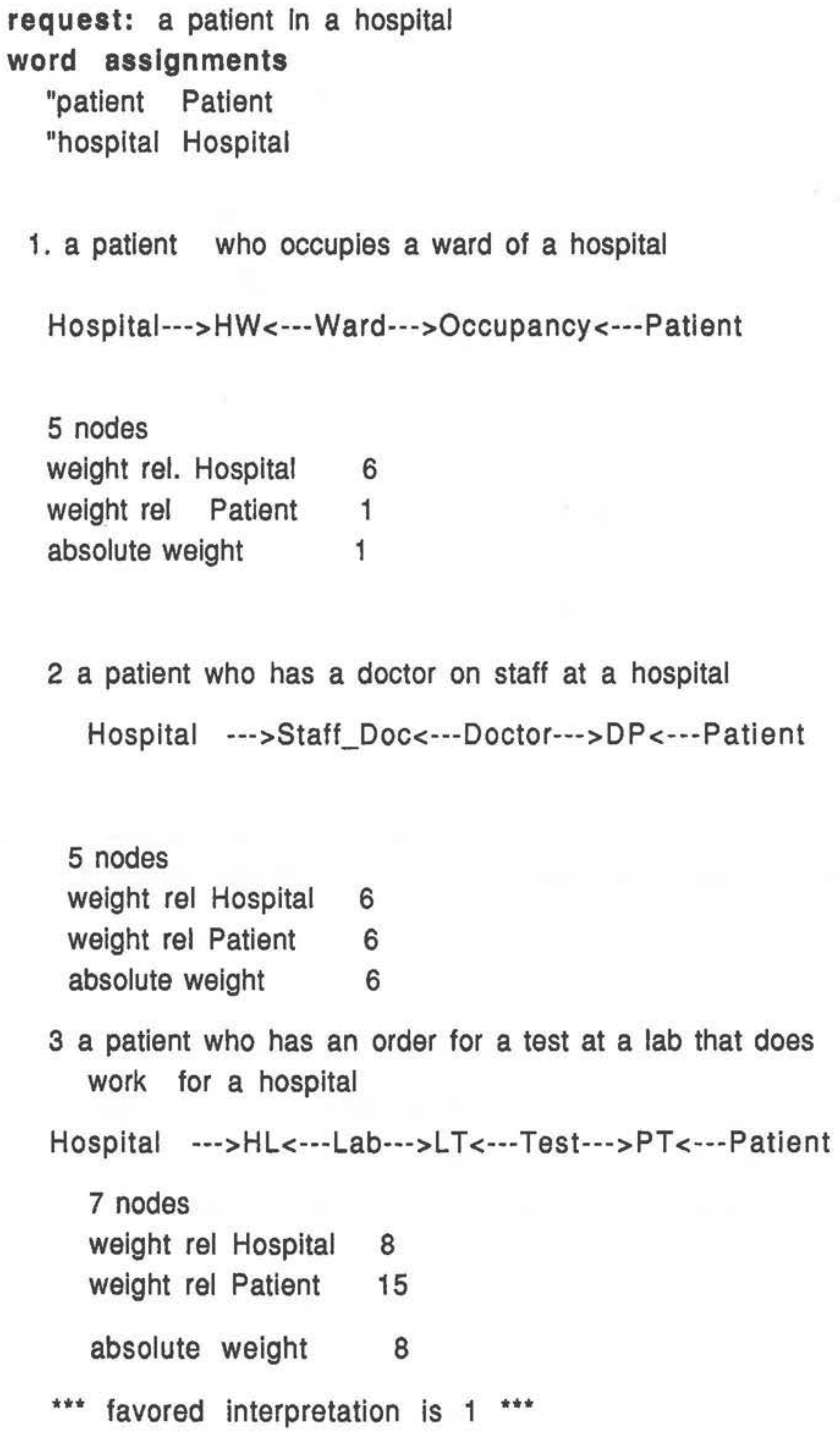
Staff_Doc - associates with a hospital those doctors that are on staff at the hospital

HL - associates with a hospital those labs that are doing work for the hospital

LT - associates with a lab the medical tests that are to be performed at the lab

PT - associates with a patient those medical test that have been ordered for the patient

Figure 9: Domain Graph for the Hospital Domain



```
request:  a patient in a hospital
word  assignments
   "patient   Patient
   "hospital  Hospital

  1. a patient   who occupies a ward of a hospital

   Hospital--->HW<---Ward--->Occupancy<---Patient


   5 nodes
   weight rel. Hospital      6
   weight rel    Patient     1
   absolute weight           1


   2 a patient who has a doctor on staff at a hospital

      Hospital   --->Staff_Doc<---Doctor--->DP<---Patient


    5 nodes
    weight rel Hospital     6
    weight rel Patient      6
    absolute weight         6

   3 a patient who has an order for a test at a lab that does
      work   for a hospital

   Hospital   --->HL<---Lab--->LT<---Test--->PT<---Patient

      7 nodes
      weight rel Hospital    8
      weight rel Patient     15

      absolute  weight        8

   ***  favored  interpretation  is  1  ***
```

Figure 10: Resolution of Semantic Ambiguity in “a patient in a hospital”

```
request:  a patient in a hospital with  tests
word  assignments
  "patient"  Patient      "hospital"  Hospital       "test"     Test

 1. a patient who occupies a bed in a ward of a hospital and for whom
    tests have been ordered

  Hospital--->HW<---Ward--->Occupancy<---Patient--->PT<---Test

  7 nodes

  weight rel.  Hospital    15           weight rel   Test          1
  weight rel   Patient     8            weight rel   PT            1

  absolute weight          1


 2.  a patient who occupies a ward of a hospital which has orders for
    tests at some lab

  nodes 10

  Hospital   --->HL<---Lab--->LT<---Test--->Hosp_Test
     |
     --->HW<---Ward--->Occupancy<---Patient

  weight rel  Hospital  32       weight rel Patient  22

  weight rel  Test  31           weight rel  HT        31

  absolute weight   22

 Hospital   --->HL<---Lab--->LT<---Test
    |   |
    |   --->Hosp_Test
    |
    --->HW<---Ward--->Occupancy<---Patient

  weight rel Hospital    32       weight rel Test   31
  weight rel Patient     21       weight rel Hosp_Test  31
  absolute  weight       21

  ***  favored  interpretation  is  1  ***
```

Figure 11: Attachment of Modifiers in "a patient in a hospital with tests"

request: Dr. Lee's students

word assignments

"Dr. Lee" PName

"student" Student

1. Students in courses taught by Dr. Lee

```
Student--->SC<---Course--->CP<---Prof--->PN<---PName
       (0,n)  (0,n)     (1,n)  (0,n)    (1,1)   (0,n)
```

```
7 nodes
weight rel. Student  (X+Z+0)=X+Z
weight rel PName     (X+X+X)=3X
absolute weight      minimum(X+Z,3X)=X+Z
```

2. Students supervised by Dr. Lee

```
Student  --->Sup<---Prof--->PN<---Pname
          (1,1)    (0,n)   (1,1)  (0,n)
```

```
5 nodes
weight rel Student   (0+0)=0
weight rel PName     (X+X)=2X
absolute weight      minimum(0,2X)=0
```

Figure 12: Varying the Parameters for "Dr. Lee's students"

References

[1] Hiyan Alshawi and Robert C. Moore. *Feasibility Study for a Research Programme in Natural-Language Processing.* SRI International Cambridge Computer Science Research Center, Cambridge, England, August 1986. SRI Project ECC-1437. Contract No. ALV/CONS/IKBS/026.

[2] P.A. Bernstein and C.W. Chiu. Using semijoins to solve relational queries. *J. ACM*, 28(1):25–40, 1981.

[3] P.P. Chen. The Entity-Relationship model - toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, 1976.

[4] Bipin C. Desai, Richard J. Pollock, and Philip J. Vincent. A natural language interface to a multiple databased office information system. *SIGOIS Bulletin*, 9(4):19–33, 1988.

[5] Charles J. Fillmore. The case for case. In E. Bach and R.T. Harms, editors, *Universals in Linguistics*, pages 1–88, New York, 1968. Holt, Rinehart, and Winston.

[6] Paul C. Gilmore. Natural deduction based set theories: A new resolution of the old paradoxes. *J. Symb. Logic*, 51(2), 1986.

[7] Paul C. Gilmore. Concepts and methods for database design. Technical Report TR87-31, Department of Computer Science, University of British Columbia, 1987.

[8] Paul C. Gilmore. A foundation for the Entity Relationship approach: How and why? In *Proceedings of the 7th International Conference on Entity-Relationship Approach*, New York, 1987.

[9] Barbara J. Grosz, Douglas E. Appelt, Paul A. Martin, and Fernando C.N. Pereira. Team: An experiment in the design of transportable natural-language interfaces. In *Artificial Intelligence*, volume 32, pages 173–243. Elsevier Science Publishers B.V. (North-Holland), 1987.

[10] Gary Hall, WoShun Luk, and Nick Cercone. Disambiguating queries using dependency graphs. Technical Report LCCR TR 87-7, School of Computing Science, Simon Fraser University, Burnaby, British Columbia, Canada, 1987.

[11] G. G. Hendrix, E. D. Sacerdoti, D. Sagalowicz, and J. Slocum. Developing a natural language interface to complex data. *ACM Transactions on Database Systems*, 3(2):105–147, 1978.

[12] Jurgen M. Janas. The semantics-based natural language interface to relational databases. In L. Bolc and M. Jarke, editors, *Topics in Information Systems, Cooperative Interfaces to Information Systems*, pages 143–188. Springer-Verlag, 1986.

[13] Karen Jensen and Jean-Louis Binot. Disambiguating prepositional phrase attachments by using on-line dictionary definitions. *Computational Linguistics*, 13(3-4):251–260, 1987.

[14] Julia A. Johnson. A data management strategy for transportable natural language interfaces. Technical Report TR89-23, University of British Columbia, Dept. of Computer Science, Vancouver, B.C. Canada, 1989.

[15] Candace E. Kalish and Matthew B. Cox. Porting an extensible natural language interface: A case history. In *Proceedings of AAAI*, pages 556–560, 1987.

[16] S. Jerrold Kaplan. Designing a portable natural language database query system. *ACM Transactions on Database Systems*, 9(1):1–19, March 1984.

[17] W. Kent. Fact-based data analysis and design. In C.G. Davis, S. Jajodia, P.A. Ng, and R.T.Yeh, editors, *Entity-Relationship Approach to Software Engineering*, pages 3–53. Elsevier Science Publishers B.V. North-Holland, 1983.

[18] Veda C. Storey and Robert C. Goldstein. A methodology for creating user views in database design. *ACM Transactions on Database Systems*, 13(3):305–338, 1988.

[19] D.C. Tsichritzis and F.H. Lochovsky. *Data Models.* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1982.

[20] Joseph A. Wald and Paul G. Sorenson. Resolving the query inference problem using steiner trees. *ACM Transactions on Database Systems*, 9(3):348–368, September 1984.

[21] Joseph A. Wald and Paul G. Sorenson. Explaining ambiguity in a formal query language. *ACM Transactions on Database Systems*, 1989. To appear.

[22] Steven John White. A portable natural language database query system. Master's thesis, University of British Columbia, Department of Computer Science, Vancouver, B.C., Canada, 1985.

[23] William A. Woods. Knowledge representation. In Thomas C. Bartee, editor, *Expert Systems and Artificial Intelligence - Applications and Management*, pages 147–176. Howard W. Sams and Company, 1988.