(4)

BBN Systems and Technologies Corporation

A Subsidiary of Bolt Beranek and Newman Inc.

BBN Report No. 6463
Contract No. N00014-85-C-0016

RESEARCH AND DEVELOPMENT IN NATURAL LANGUAGE UNDERSTANDING AS PART OF THE STRATEGIC COMPUTING PROGRAM

Annual Report
21 December 1986-20 December 1987

Ralph M. Weischedel

Prepared by:

BBN Systems and Technologies Corporation
10 Moulton Street
Cambridge, MA 02138

Prepared to:

Defense Advanced Research Projects Agency (DARPA)
1400 Wilson Blvd.
Arlington, VA 22209

ARPA Order No. 5257 Contract No. N00014-85-C-0016

Scientific Officer:
Dr. Alan R. Meyrowitz

BBN Report No. 6463
Contract No. N00014-85-C-0016

RESEARCH AND DEVELOPMENT IN NATURAL LANGUAGE UNDERSTANDING AS PART OF THE STRATEGIC COMPUTING PROGRAM

Annual Report
21 December 1986-20 December 1987

Ralph M. Weischedel

Prepared by:

BBN Systems and Technologies Corporation
10 Moulton Street
Cambridge, MA 02138

Prepared to:

Defense Advanced Research Projects Agency (DARPA)
1400 Wilson Blvd.
Arlington, VA 22209

ARPA Order No. 5257 Contract No. N00014-85-C-0016

Scientific Officer:
Dr. Alan R. Meyrowitz

BBN Systems and Technologies Corporation Report No. 6463

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by ONR under Contract No. 00014-85-C-0016. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

ECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
6463

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
BBN Systems and Technologies
Corporation

6b. OFFICE SYMBOL
(If applicable)

7a. NAME OF MONITORING ORGANIZATION
Office of Naval Research

6c. ADDRESS (City, State, and ZIP Code)
10 Moulton Street
Cambridge, MA 02138

7b. ADDRESS (City, State, and ZIP Code)
Department of the Navy
Arlington, VA 22217

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Defense Advanced Research
Projects Agency

8b. OFFICE SYMBOL
(If applicable)
DARPA ISTO

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)
1400 Wilson Blvd.
Arlington, VA 22209

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
Research and Development in Natural Language Understanding as Part of the Strategic Computing Program Annual Report 21 December 1986-20 December 1987

12. PERSONAL AUTHOR(S)

13a. TYPE OF REPORT

13b. TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT (Year, Month, Day)

15. PAGE COUNT
54

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Hybrid representation; ill-formed input, knowledge representation, natural language processing, IRUS, Janus.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

In this report two papers stemming from work dating from 1987 are reprinted In the first, we report pioneering work in knowledge representation for natural language processing.

In BBN's natural language understanding and generation system (Janus), we have used a hybrid approach to representation, employing an intensional logic for the representation of the semantics of utterances and a taxonomic language with formal semantics for specification of descriptive constants and axioms relating them. Remarkably, 99.9% of 7,000 vocabulary items in our natural language applications could be adequately axiomatized in the taxonomic language.

In the second paper, we overview the problems, issues, and approaches to dealing with ill-formed input.

cont. on back side...

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL

22b. TELEPHONE (Include Area Code)

22c. OFFICE SYMBOL

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Box 19 cont.

There are several purposes of this paper:

1. to define a class of phenomena that may be termed ill-formed input,

2. to differentiate this class from problems of spoken language understanding,

3. to briefly summarize the state of the art in understanding ill-formed input,

4. to indicate areas for further theoretical progress, and

5. to speculate regarding where there may be strong practical pay-off.

Table of Contents

List of Figures

1. A Hybrid Approach To Representation in Janus

Ralph M. Weischedel[1]

Abstract

In BBN's natural language understanding and generation system (Janus), we have used a hybrid approach to representation, employing an intensional logic for the representation of the semantics of utterances and a taxonomic language with formal semantics for specification of descriptive constants and axioms relating them. Remarkably, 99.9% of 7,000 vocabulary items in our natural language applications could be adequately axiomatized in the taxonomic language.

1.1 Introduction

Hybrid representation systems have been explored before [9, 27, 34], but until now only one has been used in an extensive natural language processing system. KL-TWO [34], based on a propositional logic, was at the core of the mapping from formulae to lexical items in the Penman generation system [31]. In this paper we report some of the design decisions made in creating a hybrid of an intensional logic with a taxonomic language for use in Janus, BBN's natural language system, consisting of the IRUS-II understanding components [5] and the Spokesman generation components [19, 22]. To our knowledge, this is the first hybrid approach using an intensional logic, and the first time a hybrid representation system has been used for understanding.

In Janus, the meaning of an utterance is represented as an expression in WML (World Model Language) [15], which is an intensional logic. However, a logic merely prescribes the framework of semantics and of ontology. The *descriptive constants*, that is, the individual constants (functions with no arguments), the other function symbols, and the predicate symbols, are abstractions without any detailed commitment to ontology. (We will abbreviate *descriptive constants* throughout the remainder of this paper as *constants*.)

[1] A modified version of this paper appears in the *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, 26-29 June 1989, University of British Columbia, Vancouver, BC Canada. Requests for copies should be addressed to:

Dr. Danald E. Walker (ACL)
Bell Communications Research
435 South Street MRE 2A379
Morristown, NJ 07960, USA

Axioms stating the relationships between the constants are defined in NIKL [8, 24]. We wished to explore whether a language with limited expressive power but fast reasoning procedures is adequate for core problems in natural language processing. The NIKL axioms constrain the set of possible models for the logic in a given domain.

Though we have found clear examples that argue for more expressive power than NIKL provides, 99.9% of the examples in our expert system and data base applications have fit well within the constraints of NIKL. Based on our experience and that of others, the axioms and limited inference algorithms can be used for classes of anaphora resolution, interpretation of highly polysemous or vague words such as have and with finding omitted relations in novel nominal compounds, and selecting modifier attachment based on selection restrictions.

Sections 1.2 and 1.3 describe the rationale for our choices in creating this hybrid. Section 1.4 illustrates how the hybrid is used in Janus. Section 1.5 briefly summarizes some experience with domain-independent abstractions for organizing constants of the domain. Section 1.6 identifies related hybrids, and Section 1.7 summarizes our conclusions.

1.2 Commitments to Component Representation Formalisms

We chose well-documented representation languages in order to focus on formally specifying domains and using that specification in language processing rather than on defining new domain-independent representation languages.

A critical decision was our selection of intensional logic as the semantic representation language. (Our motivations for that choice are covered in Section 1.2.1.) Given an intensional logic, the fundamental question was how to support inference for semantic and discourse processing. The novel aspect of the design was selecting a taxonomic language and associated inference techniques for that purpose.

1.2.1 Why an Intensional Logic

First and foremost, though we had found first-order representations adequate (and desirable) for NL interfaces to relational data bases, we felt a richer semantic representation was important for future applications. The following classes of representation challenges motivated our choice.

- Explicit representations of time and world. Object-oriented simulation systems were an application that involved these, as were expert systems supporting hypothetical worlds.

The underlying application systems involved a tree of possible worlds. Typical questions about these included *What if the stop time were 20 hours?* to set up a possible world and run a simulation, and *In which situations is blue attrition greater than 50%?* where the whole tree of worlds is to be examined. The potential of time-varying entities existed in some of the applications as well, whether attribute values (as in *How often has USS Enterprise been C3?*) or entities (*When was CV22 decommissioned?*) The time and world indices of WML provided the opportunity to address such semantic phenomena (though a modal temporal logic or other logics might serve this prupose).

- Distributive/collective quantification. Collective readings could arise, though they appear rare, e.g., *Do USS Frederick's capabilities include anti-submarine warfare* or *When did the ships collide?* See [28] for a computational treatment of distributive/collective readings in WML.
- Generics and Mass Terms. Mass terms and generally true statements arise in these applications, such as in *Do nuclear carriers carry JP5?*, where JP5 is a kind of jet fuel. Term-forming operators and operators on predicates are one approach and can be accommodated in intensional logics.
- Propositional Attitudes. Statements of user preference, e.g., *I want to leave in the afternoon*, should be accommodated in interfaces to expert systems, as should statements of belief, *I believe I must fly with a U.S. carrier*. Since intensional logics allow operators on predicates and on propositions, such statements may be conveniently represented.

Our second motivation for choosing intensional logic was our desire to capitalize on other advantages we perceived for applying it to natural language processing (NLP), such as the potential simplicity and compositionality of mapping from syntactic form to semantic representation and the many studies in linguistic semantics that assume some form of intensional logic.

However, the disadvantages of intensional logic for NLP include:

- The complexity of logical expressions is great even for relatively straightforward utterances using Montague grammar [23]. However, by adopting intensional logic while rejecting Montague grammar, we have made some inroads toward matching the complexity of the proposition to the complexity of the utterance; that simplicity is at the expense of using a more powerful semantic interpreter and of sacrificing compositionality in those cases where language itself appears non-compositional.
- Real-time inference strategies are a challenge for so rich a logic. However, our hypothesis is that large classes of the linguistic examples requiring common sense reasoning can be handled using limited inference algorithms on a taxonomic language. Arguments supporting this hypothesis appear in [2, 13] for interpreting nominal compounds; in [6, 7, 32], for common sense reasoning about modifier attachment; and in [35] for phenomena in definite reference resolution.

This second disadvantage, the goal of tractable, real-time inference strategies, is the basis for adding taxonomic reasoning to WML, giving a hybrid representation.

1.2.2 Why a Taxonomic Language

Our hypothesis is that much of the reasoning needed in semantic processing can be supported by a taxonomy. The ability to pre-compile pre-specified inferential chains, to index them via concept name and role name, and to employ taxonomic inheritance for organizing knowledge were critical in selecting taxonomic representation to supplement WML.

The well-defined semantics of NIKL was the basis for choosing it over other taxonomic systems. A further benefit in choosing NIKL is the availability of KREME [1], which can be used as a sophisticated browsing, editing, and maintenance environment for taxonomies such as those written in NIKL; KREME has proven effective in a number of BBN expert system efforts other than NLP and having a taxonomic knowledge base.

In choosing NIKL to axiomatize the constants, one could use its built-in, incomplete inference algorithm, the classifier [30]. In Janus, the classifier is used only for consistency checking when modifying or loading the taxonomic network; any concepts or roles identified by the classifier as identical are candidates for further axiomatization. Our semantic procedures do not need even as sophisticated an algorithm as the NIKL classifier; pre-compiled, pre-defined inference chains in the network are simpler, faster, and have proven adequate for NLP in our applications.

1.2.3 Two Critical Choices in the Hybrid

1.2.3.1 Representing Predicates of Arbitrary Arity

Choosing a taxonomic language, at least in current implementations, means that one is restricted to unary and binary predicates. However, this is not a limitation in expressive power. One can represent a predicate P of n arguments via a unary predicate P' and n binary predicates, which is what we have done. (P r1, ..., rn) will be true iff the following expression is.

$$(\exists b)\,(\wedge\,(P'\ b)\,(R1\ b\ r1)\,(R2\ b\ r2)\ \ldots\ (Rn\ b\ rn))$$

Davidson [5] has argued for such a representation of processes on semantic grounds, since many event descriptors appear with a variable number of arguments.

1.2.3.2 Time and World Indices

Any concept name or role name in the network is a constant in the logical language. We use concepts only to represent sets of entities indexed by time and world. Roles are used only to represent sets of pairs of entities, i.e., binary relations. Given time and world indices potentially on each constant in WML, we must first state the role those indices play in the NIKL portion of the hybrid.

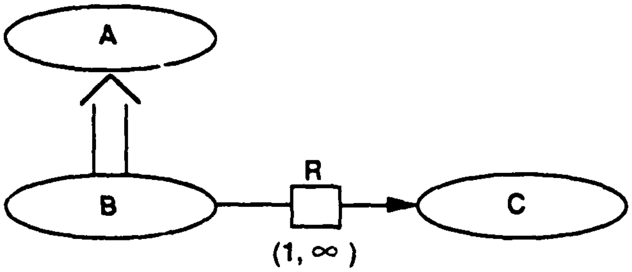

Figure 1-1: Two Typical Facts Stated in NIKL

In a first-order extensional logic, the normal semantics of SUPERC and of roles in NIKL are well defined [29]. For instance, the diagram in figure 1-1 would mean

$(\forall x)((B\ x) \supset (A\ x))$

$(\forall x)((B\ x) \supset (\exists y)(\wedge(C\ y)\ (R\ x\ y)))$.

Due to a suggestion by David Stallard, we have chosen to interpret SUPERC and the role link similarly, *but interpreted under modal necessity*, i.e., as propositions true at all times in all worlds. Thus in the diagram in Figure 1-1, (A z), (B z), (C z), and (R x y) are intensions, i.e., functions with arguments of time and world [t, w] to extensions. Rewriting the axioms above by quantifying over all times and worlds, the axioms for the diagram in Figure 1-1 in the hybrid representation are

$\Box\ (\forall x)((B\ x) \supset (A\ x))$

$\Box\ (\forall x)((B\ x) \supset (\exists y)(\wedge\ (C\ y)\ (R\ x\ y)))$.

Though this handles the overwhelming majority of constants we need to axiomatize, it does not allow for representing constants taking intensional arguments because the axioms above allow for quantification over extensions only.[2] The semantics of predicates which should have intensions as arguments are unfortunately specified separately. Examples that have arisen in our applications involve changes in a reading on a scale, e.g., *USS Stark's readiness downgraded from C1 to C4*.[3] We would like to treat that sentence as:

$(\wedge\ (\mathit{DOWNGRADE}\ a)$
$\quad (\mathit{SCALE}\ a\ (\mathit{INTENSION}\ \mathit{Stark\text{–}readiness}))$
$\quad (\mathit{PREVIOUS}\ a\ C1)$
$\quad (\mathit{NEW}\ a\ C4))$.

That is, for the example we would like to treat the scale as intensional, but have no way to do so in NIKL. Therefore, we had to annotate the definition of *downgrade* outside of the formal

[2] It is possible that one could extend NIKL semantics to allow for intensional arguments, but this has not been done.

[3] An analogy in more common terminology would be *His temperature dropped from 104 degrees to 99 degrees*.

semantics of NIKL. Only 0.1% of the 7,000 (root) word vocabulary in our applications could not be handled with NIKL. (The additional problematic vocabulary were *upgrade*, *project*, *report*, *change*, and *expect*.)

1.3 Example Representational Decisions

Here we mention some of the issues we focussed on in developing Janus. The specification of WML appears in [15]; specifications for NIKL appear in [24, 29].

Few constants. One decision was to use as few constants as possible, deriving as many entities as possible using operators in the intensional logic. In this section we illustrate this point by showing how definitely referenced sets, information about kinds, indefinitely identified sets, and generic information can be stated by derivation from a single constant whose extension is the set of all individuals of a particular class.

Some of the expressive power of the hybrid is illustrated below as it pertains to minimizing the constants needed. From the constants BLACK-ENTITIES, GRAY-ENTITIES, CATS and MICE, the operators THE, POWER, KIND, and SAMPLE are used to derive the entities corresponding to definite sets, generic classes, and indefinite sets. In a semantic network without the hybrid, one might choose (or need) to represent each of our derived entities by a node in the network. Our use of the operator THE, and the operator POWER for definite plurals follows Scha [28]. The operators KIND and SAMPLE follow Carlson's analysis [10] of the semantics of bare plurals.

THE, as an operator, takes three arguments: a variable, a sort (unary predicate), and a proposition. Its denotation is the unique salient object in context such that it is in the sort and such that if the variable is bound to it, the proposition is true. POWER takes a sort as argument and produces the predicate corresponding to the power set of the set denoted by the sort. These operators are useful for representing definite plurals; *the black cats* would be represented as (THE x (POWER CATS) (BLACK-ENTITIES x)).

SAMPLE takes the same arguments as THE, but indicates some set of entities satisfying the sort and proposition, not necessarily the largest set. KIND takes a sort as argument, and produces an individual representing the sort; its only use is for bare plurals that are surface subjects of a generic statement. If we are predicating something of a bare plural, KIND is used; for instance, *cats* as in *cats are ferocious* is represented as (KIND CATS). An indefinite set arising as a bare plural in a VP is represented using SAMPLE; for instance, *gray mice* as in *Cats eat gray mice* is represented as (SAMPLE x MICE (GRAY-ENTITIES x)).

The examples above demonstrate that an intensional logic enables derivation of many entities from fewer constants than would be needed in NIKL or other frame-based systems. The next example illustrates how the intensional logic lets us express some propositions that can be stated in many semantic network systems, but not in NIKL.

Generic assertions. Generic statements such as *Cats eat mice* are often encoded in a semantic network or frame system. This is not possible in the semantics of NIKL, but is possible in the hybrid. The structure in Figure 1-2 would not give the desired generic meaning, but rather would mean (ignoring time and world) that

$$(\forall x)\,((\mathit{CATS}\ x) \supset (\exists y)(\wedge\ (\mathit{MICE}\ y)(\mathit{EAT}\ x\ y))),$$

i.e., every cat eats some mouse.

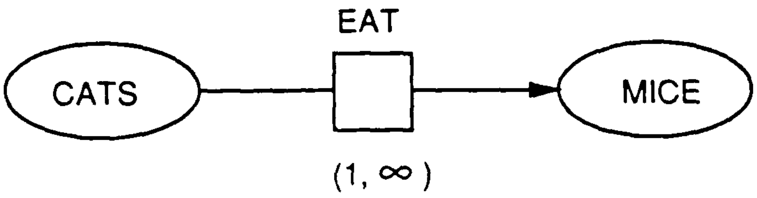

Figure 1-2: Illustration Distinguishing NIKL Networks from other Semantic Nets

Again, following Carlson's linguistic analysis [10], in the hybrid we would have a generic statement about the kind corresponding to cats, that these eat indefinitely specified sets of mice. GENERIC is an operator which produces a predicate on kinds, intuitively meaning that the resulting predicate is typically true of individuals of the kind that is its argument. Our formal representation (ignoring tense for simplicity) is

$$(\mathit{GENERIC}\ (\mathit{LAMBDA}\ (x)\ (\mathit{EAT}\ x\ (\mathit{SAMPLE}\ y\ \mathit{MICE}))))\ (\mathit{KIND}\ \mathit{CATS}).$$

Next we illustrate a potential powerful feature of the hybrid which we have chosen not to exploit.

Derivable definitions. The hybrid gives a powerful means of defining lexical items. To define *pilot*, one wants a predicate defining the set of people that typically are the actors in a flight, i.e.,

$$\begin{array}{l}(\mathit{LAMBDA}\ (x')\ \{\wedge\ (\mathit{PERSON}\ x') \\ \quad (\mathit{GENERIC}\ (\mathit{LAMBDA}\ (x)\ (\exists y)(\wedge\ (\mathit{FLYING\text{-}EVENT}\ y)\ (\mathit{ACTOR}\ y\ x))))\ x')\})\end{array}$$

Though the hybrid gives us the representational capacity to make such definitions, we have chosen as part of our design not to use it. For to use it, would mean stepping outside of NIKL to specify constants, and therefore, that the reasoning algorithms based on taxonomic semantics would not be the simple, efficient strategies, but rather might require arbitrarily complex theorem proving for expressions in intensional logic.[4]

1.4 Use of the Taxonomy in Janus

By *domain model* we mean the set of axioms encoded in NIKL regarding the constants. The domain model serves several purposes in Janus. Of course, in defining the constants of our semantic representation language, it provides the constants that can appear in formulae that lexical items map to. For instance, *vessel* and *ship* map to VESSEL. In the example above regarding *pilot*, the constants were PERSON, FLYING-EVENT, and ACTOR; in the formula above stating that cats eat mice, the constants were EAT, MICE, and CATS.

In this section, we divide the discussion in three parts: current uses of the domain model in Janus; a plausible, but rejected use; and proposals for its use, but not yet implemented.

1.4.1 Current Uses

1.4.1.1 Selection Restrictions

The domain model provides the semantic classes (or sorts of a sorted logic) that form the primitives for selection restrictions. Its use for this purpose is neither novel nor surprising, merely illustrative. In the case of *deploy*, a MILITARY-UNIT can be the logical subject, and the object of a phrase marked by *to* must be a LOCATION. Almost all selection restrictions are based on the semantic class of the entities described by a noun phrase. That is, almost all may be checked by using taxonomic knowledge regarding constants.

A table of semantic classes for the operators discussed earlier is provided in Figure 1-3. Though the logical form for *the carriers*, *all carriers*, *some carriers*, *a carrier*, and *carriers* (both in the KIND and SAMPLE case) varies, the selection restriction must check the NIKL network for consistency between the constant CARRIERS and the constraint of the selection restriction. To see this, consider the case of *command* (in the sense of a military command) which requires that its direct object in active clauses be a MILITARY-UNIT and that its surface subject in passive clauses be a MILITARY-UNIT, i.e., its logical object must be a MILITARY-UNIT. Suppose *USS Enterprise*, *carrier*, and *aircraft carrier* all have semantic class CARRIER. Since an ancestor of CARRIER in the taxonomy is MILITARY-UNIT, each of those phrases

[4] USC/ISI [20] has proposed a first-order formula defining the set of items that have ever been the actor in a flight. Their definition is solely within NIKL using the QUA link [14], which is exactly the set of fillers of a slot. While having ever flown could be a sense of *pilot*, it seems less useful than the sense of normally flying a plane.

satisfy the aforementioned selection restriction on the verb *command*. Phrases whose class does not have MILITARY-UNIT as an ancestor or as a descendent[5] will not satisfy the selection restriction. That is, definite evidence of consistency with the selection restriction is normally required.

Expression	Semantic Class
(THE x P (R x))	P
(POWER P)	P
(KIND P)	P
(SAMPLE x P (R x))	P
(LAMBDA x P (R x))	P

Figure 1-3: Relating Expressions to Classes[6]

There are three cases where more must be done. For pronouns, Janus saves selection restrictions that would apply to the pronoun's referent, later applying those constraints to eliminate candidate referents. Metonymy is an exception, discussed in Section 1.4.3.2. There are cases of selection restrictions requiring information additional to the semantic class, but these are checked against the type of the logical expression[7] for a noun phrase, rather than its semantic class only. *Collide* requires a set of agents. The type of a plural, for instance, is (SET P), where P is its semantic class. The selection restriction on *collide* could be represented as (SET PHYSICAL-OBJECT).

1.4.1.2 Knowledge Acquisition

We have developed two complementary tools to greatly increase our productivity in porting BBN's Janus NL understanding and generation system to new domains. IRACQ [3] supports learning lexical semantics from examples with only one unknown word. IRACQ is used for acquiring the diverse, complex patterns of syntax and semantics arising from verbs, by providing examples of the verb's usage. Since IRACQ assumes that a large vocabulary is available for use in the training examples, a way to rapidly infer the knowledge bases for the overwhelming majority of words is an invaluable complement.

KNACQ [36] serves that purpose. The domain model is used to organize, guide, and assist in acquiring the syntax and semantics of domain-specific vocabulary. Using the browsing facilities, graphical views, and consistency checker of KREME [1] on NIKL taxonomies, one

[5] We check whether the constraint is a descendent of the class of the noun phrase to determine whether consistency is possible. For instance, if *decomission* requires a VESSEL as the object of the decommisioning, *those units* and *they* satisfy the selection constraint.

[6] The rules may need to be used recursively to get to a constant.

[7] Every expression in WML has a type.

may select any concept or role for knowledge acquisition. KNACQ presents the user with a few questions and menus to elicit the English expressions used to refer to that concept or role.

To illustrate the kinds of information that must be acquired consider the examples in Figure 1-4.

The vessel speed of Vinson
The vessel's speed is 5 knots
Its speed
Which vessels are deployed C3?
The vessels with speed above 20 knots
Vinson has speed less than 20 knots
Which vessels have a CROVL of C3?

Figure 1-4: Examples for Knowledge Acquisition

To handle these one would have to acquire information on lexical syntax, lexical semantics, and mapping to expert system structure for all words not in the domain-independent dictionary. For purposes of this exposition, assume that the words, *vessel, speed, Vinson, CROVL, C3,* and *deploy* are to be defined. *A vessel has a speed of 20 knots* or *a vessel's speed is 20 knots* would be understood from domain-independent semantic rules regarding *have* and *be*, once lexical information for *vessel* and *speed* is acquired. In acquiring the definitions of *vessel* and *speed*, the system should infer interpretations for phrases such as *the speed of a vessel, the vessel's speed,* and *the vessel speed.*

Given the current implementation, the required knowledge for the words *vessel, speed,* and *CROVL* is most efficiently acquired using KNACQ; names of instances of classes, such as *Vinson* and *C3* are automatically inferred from instances; and knowledge about *deploy* and its derivatives would be acquired via IRACQ.

To illustrate this acquistion centered around the domain model, consider acquistion centered around roles. *Attributes* are binary relations on classes that can be phrased as *the <relation> of a <class>*. For instance, suppose CURRENT-SPEED is a binary relation relating vessels to SPEED, a subclass of ONE-D-MEASUREMENT. An attribute treatment is the most appropriate, for *the speed of a vessel* makes perfect sense. KNACQ asks the user for one or more English phrases associated with this functional role; the user response in this case is *speed*. That answer is sufficient to enable the system to understand the kernel noun-phrases listed in Figure 1-5. Since ONE-D-MEASUREMENT is the range of the relation, the software knows that statistical operations such as average and maximum apply to speed. The lexical information inferred is used compositionally with the syntactic rules, domain independent semantic rules, and other lexical semantic rules. Therefore, the generative capacity of the lexical semantic and syntactic information is linguistically very great, as one would expect. A small subset of the examples illustrating this without introducing new domain specific lexical items appears in Figure 1-5.

KERNEL NOUN PHRASES

the speed of a vessel
the vessel's speed
the vessel speed

RESULTS from COMPOSITIONALITY

The vessel speed of Vinson
The vessels with a speed of 20 knots
Vinson has speed less than 20 knots
Its speed
Which vessels have speeds
Carriers with speed 20 knots
Vinson has speed 1
The vessel's speed is 5 knots
Their greatest speed
Which vessels have speed above 20 knots
Eisenhower has Vinson's speed
Their average speeds

Figure 1-5: Attribute Examples

Some lexicalizations of roles do not fall within the attribute category. For these, a more general class of regularities is captured by the notion of caseframe rules. Suppose we have a role UNIT-OF, relating CASREP and MILITARY-UNIT. KNACQ asks the user which subset of the following six patterns in Figure 1-6 are appropriate plus the prepositions that are appropriate.

1. <CASREP> is <PREP> <MILITARY-UNIT>
2. <CASREP> <PREP> <MILITARY-UNIT>
3. <MILITARY-UNIT> <CASREP>
4. <MILITARY-UNIT> is <PREP> <CASREP>
5. <MILITARY-UNIT> <PREP> <CASREP>
6. <CASREP> <MILITARY-UNIT>

Figure 1-6: Patterns for the Caseframe Rules

For this example, the user would select patterns (1), (2), and (3) and select *for*, *on*, and *of* as prepositions.[8]

The information acquired through KNACQ is used both by the understanding components and by BBN's Spokesman generation components for paraphrasing, for providing clarification responses, and for answers in English. Mapping from the WML structures to lexical items is accomplished using rules acquired with KNACQ, as well as handcrafted mapping rules for lexical items not directly associated with concepts or roles.

[8] Normally, if pattern (1) is valid, pattern (2) will be as well and vice versa. Similarly, if pattern (4) is valid, pattern (5) will normally be also. As a result, the menu items are coupled by default (selecting (1) automatically selects (2) and vice versa), but this default may be simply overridden by selecting either and then deselecting the other. The most frequent examples where one does not have the coupling of those patterns is the preposition *of*.

1.4.1.3 Highly Polysemous Words

Have, with, and *of,* are highly polysemous. Some of their senses are very specific, frozen, and predictable, e.g., *to have a cold*; these senses may be itemized in the lexicon. However, other senses are vague, if considered in a domain-independent way; nevertheless, they must be resolved to precise meanings if accessing a data base, expert system, etc. *USS Frederick has a speed of 30 knots* has this flavor, for the general sense is associating an attribute with an entity.

To handle such cases, we look for a relation R in the domain model which could be the domain-dependent interpretation. If *A has B, the B of A,* or *A with B* are input, the semantic interpreter looks for a role R from the class associated with A to the class associated with B. If no such role exists, the search is for a role relating the nearest ancestor of the class of A to any ancestor of the class of B. The implicit assumption is that items structured closely together in the domain model can be related with such vague words, and that items that can be related via such vague words will naturally have been organized closely together in the domain model.

While describing the procedure as a search, in fact, an explicit run-time search may not be necessary. All SUPERCs (ancestors) of a concept are compiled and stored when the taxonomy is loaded. All roles from one concept to another are also pre-compiled and stored, maintaining the distinction between roles that are explicit locally versus those that are compiled. Furthermore, the ancestors and role relations are indexed. One need only walk up the chain of ancestors if no locally defined role relates the two concepts, but some inherited (not locally defined) role does; then one walks up the ancestor chain(s) only to find the closest applicable role. Thus, in many cases, ''semantic reasoning'' is reduced to efficient table lookup.

1.4.1.4 Relation to Underlying System

Adopting WML offers the potential of simplifying the mapping from surface form to semantic representation, although it does increase the complexity of mapping from WML to executable code, such as SQL or expert system function calls. The mapping from intensional logic to executable code is beyond the scope of this paper; our first implementation was reported in [33]; the current implementation is described in [26].

This process makes use of a model of underlying system capabilities in which each element relates a set of domain model constants to a method for accessing the related information in the database, expert system, simulation program, etc. For example, the constant HARPOON-CAPABLE, which defines a set of vessels equipped with harpoon missiles, is associated with an underlying system model element which states how to select the subset of exactly those vessels. In a Navy relational data base that we have dealt with, the relevant code selects just those records of a table of unit characteristics with a "Y" in the HARP field.

1.4.2 Where an Alternative Mechanism was Selected

Though the domain model is central to the semantic processing of Janus, we have not used it in all possible ways, but only where there seems to be clear benefit.

In telegraphic language, omitted prepositions may arise, as in *List the creation date file B*. Alternatively, if the NLP system is part of a speech understanding system, prepositions are among the most difficult words to recognize reliably. Omitted prepositions could be treated with the same heuristic as implemented for interpreting the meaning of *have*, *with*, and *of*. However, we have chosen a different inference technique for omitted prepositions.

Though one could represent selection restrictions directly in a taxonomy (as reported in [7, 32]), selection restrictions in Janus are stored separately, indexed by the semantic class of the head word. We believe it more likely that Janus will have the selectional pattern involving the omitted preposition, than that the omitted preposition corresponds to a usage unknown to Janus and inferable from the domain model relations. Consequently, Janus applies the selection restrictions corresponding to all senses of the known head, to find what senses are consistent with the proposed phrase and with what prepositions. In practice, this gives rise to far fewer possibilities than considering all relations possible whether or not they can be expressed with a preposition.

1.4.3 Proposals not yet Implemented (Possible Future Directions)

In this section, we speculate regarding some possible future work based on further exploiting the domain model and hybrid representation system described in this paper.

1.4.3.1 An Approach to Bridging

It has long been observed [11] that mention of one class of entities in a communication can bring into the foreground other classes of entities which can be referred to though not explicitly introduced. The process of inferring the referent when such a reference occurs has been called *bridging* [12]. Some examples, taken from [12], appear below, where the reference requiring bridging is underlined.

1. *I looked into the room. The ceiling was very high.*
2. *I walked into the room. The chandeliers sparkled brightly.*
3. *I went shopping yesterday. The time I started was 3 PM.*

We believe a taxonomic domain model provides the basis for an efficient algorithm for a broad class of examples of bridging, though we do not believe that it will cover all cases. If A is the class of a discourse entity arising from previous utterances, then any entity of class B, such that the NIKL domain model has a role from A to B (or from B to A) can be referred to by a definite NP. This has not yet been integrated into the Janus model of reference processing [4].

1.4.3.2 Metonymy

Unstated relations in a communication must be inferred for full understanding of nominal compounds and metonymy. Those that can be anticipated can be built into the lexicon; the challenge is to deal with those that are novel to Janus. Finding the omitted relation in novel nominal compounds using a taxonomy has been explored and reported elsewhere [13].

We propose treating many novel cases of metonymy in the following way:

1. Where patterns of metonymy can be identified, such as using a description of a part to refer to the whole (and other patterns identified in [17]), pre-compile chains of relations between classes in the domain model, e.g., (PART-OF A B) where A and B are concepts.
2. In processing an input, when a selection restriction on an NP fails, record the failed restriction with the partial interpretation for possible future processing, after all attempts at a literal interpretation of the input have failed.
3. If no literal interpretation of the input can be found, look among the precompiled relations of step 1 above for any class that could be so related to the class of the NP that appears.
4. If a relation is applicable, attempt to resume interpretation assuming the referent of the NP is in the related class.

This has not been implemented, but offers an efficient alternative to the abductive theorem-proving approach described in [16].

1.5 Top-Level Abstractions in the NIKL Taxonomy

WML and NIKL together provide a framework for representation. The highest concepts and relations in the NIKL network provide a representational style in which more concrete constants must fit. The first abstraction structure used in Janus was the USC/ISI "upper structure" [20]. Because it seemed tied to systemic linguistics in critical ways, rather than to a more general ontological style, we have replaced it with another domain-independent set of concepts and roles. For any application domain, all domain-dependent constants must fit underneath the domain-independent structure. The domain-independent taxonomy consists of 70 concepts and 24 roles currently, but certainly could be further expanded as one attempts to further axiomatize and model notions useful in a broad class of application domains.

During the evolution of Janus, we explored whether the domain-independent taxonomy could be greatly expanded by a broad set of primitives used in the Longman Dictionary of Contemporary English [18] (LDOCE) to define domain-independent constants. LDOCE defines approximately 56,000 words in terms of a base vocabulary of roughly 2,000 items.[9] We estimate that about 20,000 concepts and roles should be defined corresponding to the 2,000 multi-way ambiguous words in the base vocabulary. The appeal, of course, is that if these basic notions were sufficient to define 56,000 words, they are generally applicable, providing a candidate for general-purpose primitives.

The course of action we followed was to build a taxonomy for all of the definitions of approximately 200 items from the base vocabulary *using the definitions of those vocabulary items themselves in the dictionary*. In this attempt, we encountered the following difficulties:

- Definitions of the base vocabulary often involved circularity.
- Definitions included assertional information and/or knowledge appropriate in defeasible reasoning, which are not fully supported by NIKL. For example, the first definition of *cat* is "a small four-legged animal with soft fur and sharp claws, often kept as a pet or for catching mice or rats."
- Multiple views and/or vague definitions and usage arose in LDOCE. For instance, the second definition of *cat* (p. 150) is "an animal *related to this* such as the lion or tiger" (italics added). Such a vague definition helped us little in axiomatizing the notion.

Thus, we decided that hand-crafted abstractions would be needed to axiomatize by hand the LDOCE base vocabulary if general-purpose primitives were to result. On the other hand, concrete concepts corresponding to a lower level of abstraction seem obtainable from LDOCE. In particular the LDOCE definitions of units of measurement for the avoirdupois and metric systems were very useful. A more detailed analysis of our experience is presented in [25].

1.6 Related Work

Several hybrid representation schemes have been created, although only ours seems to have explored a hybrid of intensional logic with an axiomatizable frame system. The most directly related efforts are the following:

- KL-TWO [34], which marries a frame system (NIKL) with propositional logic (RUP [21]). Limited inference in propositional logic is the goal of KL-TWO. Limited aspects of universal quantification are achieved via allowing demons in the inference process.

[9] Though the authors of LDOCE definitions try to stay within the base vocabulary, exceptions do arise such as diagrams and proper nouns, e.g., *Catholic Church*.

KL-TWO and its classification algorithm [30] are at the heart of the lexicalization process of the text generator Penman [31].

- KRYPTON [9], which marries a frame system with first-order logic. The frame system is designed to be less expressive than NIKL to allow rapid checking for disjointness of two class concepts in order to support efficient resolution theorem proving. KRYPTON has not as yet been used in any natural language processor.

1.7 Conclusions

Our conclusions regarding the hybrid representation approach of intensional logic plus NIKL-based axioms to define constants are based on three kinds of efforts:

- Bringing Janus up on two large expert system and data base applications within DARPA's Battle Management Programs. The combined lexicon in the effort is approximately 7,000 words (not counting morphological variations).
- The efforts synopsized in Section 1.5 towards general purpose domain notions.
- Experience in developing IRACQ and KNACQ, acquisition tools integrated with the domain model acquisition and maintenance facility KREME.

First, *a taxonomic language with a formal semantics can supplement a higher order logic in support of efficient, limited inferences needed in a natural language processor*. Based on our experience and that of others, the axioms and limited inference algorithms can be used for classes of anaphora resolution, interpretation of *have*, *with*, and *of*, finding omitted relations in novel nominal compounds, applying selection restrictions, and mapping from the semantic representation of the input to code for carrying out the user's request.

Second, *an intensional logic can supplement a taxonomic language in trying to define word senses formally*. Our effort with LDOCE definitions showed how little support is provided for defining word senses in a taxonomic language. A positive contribution of intensional logic is the ability to distinguish universal statements from generic ones from existential ones; definite sets from unspecified ones; and necessary and sufficient information from assertional information, allowing for a representation closer to the semantics of English.

Third, *the hybridization of axioms for taxonomic knowledge with an intensional logic does not allow us to represent all that we would like to, but does provide a very effective engineering approach*. Out of 7,000 lexical entries (not counting morphological variations), only 0.1% represented concepts inappropriate for the formal semantics of NIKL.

The ability to pre-compile pre-specified, inferential chains, to index them via concept name and role name, and to employ taxonomic inheritance for organizing knowledge were critical in selecting taxonomic representation to supplement WML. These techniques of pre-compiling pre-specified inferential chains and of indexing them should also be applicable to knowledge representations other than taxonomies as well.

At a later date, we hope to quantify the effectiveness of the semantic heuristics described in this paper.

Acknowledgements

This brief report represents a total team effort. Significant contributions were made by Damaris Ayuso, Rusty Bobrow, Ira Haimowitz, Erhard Hinrichs, Thomas Reinhardt, Remko Scha, David Stallard, and Cynthia Whipple. We also wish to acknowledge many discussions with William Mann and Norman Sondheimer in the early phases of the project.

References

[1] Abrett, G. and Burstein, M. The KREME Knowledge Editing Environment. *Int. J. Man-Machine Studies* 27:103-126, 1987.

[2] Ayuso Planes, D. The Logical Interpretation of Noun Compounds. Master's thesis, Massachusetts Institute of Technology, June, 1985.

[3] Ayuso, D.M., Shaked, V., and Weischedel, R.M. An Environment for Acquiring Semantic Information. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pages 32-40. ACL, 1987.

[4] Ayuso, D. Discourse Entities in Janus. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, pages 243-250. 1989.

[5] BBN Systems and Technologies Corp. *A Guide to IRUS-II Application Development in the FCCBMP*. BBN Report 6859, BBN Systems and Technologies Corp., Cambridge, MA, 1988.

[6] Bobrow, R. and Webber, B. PSI-KLONE: Parsing and Semantic Interpretation in the BBN Natural Language Understanding System. In *Proceedings of the 1980 Conference of the Canadian Society for Computational Studies of Intelligence*. CSCSI/SCEIO, May, 1980.

[7] Bobrow, R. and Webber, B. Knowledge Representation for Syntactic/Semantic Processing. In *Proceedings of the National Conference on Artificial Intelligence*. AAAI, August, 1980.

[8] Brachman, R.J. and Schmolze, J.G. An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science* 9(2), April, 1985.

[9] Brachman, R.J., Gilbert, V.P., and Levesque, H.J. An Essential Hybrid Reasoning System: Knowledge and Symbol Level Accounts of Krypton. In *Proceedings of IJCAI85*, pages 532-539. International Joint Conferences on Artificial Intelligence, Inc., Morgan Kaufmann Publishers, Inc., Los Angeles, CA, August, 1985.

[10] Carlson, G. *Reference to Kinds in English*. Garland Press, New York, 1979.

[11] Chafe, W. Discourse Structure and Human Knowledge. *Language Comprehension and the Acquisition of Knowledge*. Winston and Sons, Washington, 1972.

[12] Clark, H.H. Bridging. In *Theoretical Issues in Natural Language Processing*, pages 169-174. 1975.

[13] Finin, T.W. The Semantic Interpretation of Nominal Compounds. In *Proceedings of The First Annual National Conference on Artificial Intelligence*, pages 310-312. The American Association for Artificial Intelligence. The American Association for Artificial Intelligence, August, 1980.

[14] Freeman, M. The QUA Link. In Schmolze, J.G. and Brachman R.J. (editors), *Proceedings of the 1981 KL-ONE Workshop*, pages 55-65. Bolt Beranek and Newman Inc., 1982.

[15] Hinrichs, E.W., Ayuso, D.M., and Scha, R. The Syntax and Semantics of the JANUS Semantic Interpretation Language. In *Research and Development in Natural Language Understanding as Part of the Strategic Computing Program, Annual Technical Report December 1985 - December 1986*, pages 27-31. BBN Laboratories, Report No. 6522, 1987.

[16] Hobbs, et. al. Interpretation as Abduction. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, pages 95-103. 1988.

[17] Lakoff, G. and Johnson, M. *Metaphors We Live By*. The University of Chicago Press, Chicago, 1980.

[18] *Longman Dictionary of Contemporary English* Essex, England, 1987.

[19] MacLaughlin, D. *Parrot: The Janus Paraphraser*. BBN Report 7139, BBN Systems and Technologies Corp., Cambridge, MA, 1989.

[20] Mann, W.C., Arens, Y., Matthiessen, C., Naberschnig, S., and Sondheimer, N.K. *Janus Abstraction Structure -- Draft 2*. Technical Report, USC/Information Sciences Institute, 1985.

[21] David A. McAllester. *Reasoning Utility Package User's Manual*. AI Memo 667, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, April, 1982.

[22] Meteer, M. *The Spokesman Natural Language Generation System*. BBN Report 7090, BBN Systems and Technologies Corp., Cambridge, MA, 1989.

[23] Montague, Richard. The Proper Treatment of Quantification in Ordinary English. In J. Hintikka, J. Moravcsik and P. Suppes (editors), *Approaches to Natural Language*, pages 221-242. Reidel, Dordrecht, 1973.

[24] Moser, M.G. An Overview of NIKL, the New Implementation of KL-ONE. In Sidner, C. L., et al. (editors), *Research in Knowledge Representation for Natural Language Understanding - Annual Report, 1 September 1982 - 31 August 1983*, pages 7-26. BBN Laboratories Report No. 5421, 1983.

[25] Reinhardt, T. and Whipple, C. Summary of Conclusions from the Longman's Taxonomy Experiment. . BBN Systems and Technologies Corporation, Cambridge, MA, 1988, pages .

[26] Resnik, P. *Access to Multiple Underlying Systems in Janus*. BBN Report 7142, Bolt Beranek and Newman Inc., September, 1989.

[27] Rich, C. Knowledge Representation languages and the Predicate Calculus: How to Have Your Cake and Eat It Too. In *Proceedings of the Second National Conference on Artificial Intelligence*, pages 193-196. AAAI, August, 1982.

[28] Scha, R. and Stallard, D. Multi-level Plurals and Distributivity. In *26th Annual Meeting of the Association for Computational Linguistics*, pages 17-24. Association for Computational Linguistics, June, 1988.

[29] Schmolze, J. G., and Israel, D. J. KL-ONE. Semantics and Classification. In Sidner, C.L., et al. (editors), *Research in Knowledge Representation for Natural Language Understanding - Annual Report, 1 September 1982 - 31 August 1983*, pages 27-39. BBN Laboratories Report No. 5421, 1983.

[30] Schmolze, J.G., Lipkis, T.A. Classification in the KL-ONE Knowledge Representation System. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence.* 1983.

[31] Sondheimer, N. K. and Nebel, B. A Logical-form and Knowledge-base Design for Natural Language Generation. In *Proceedings AAAI-86 Fifth National Conference on Artificial Intelligence*, pages 612-618. The American Association for Artificial Intelligence, Morgan Kaufmann Publishers, Inc., Los Altos, CA, Aug, 1986.

[32] Sondheimer, N.K., Weischedel, R.M., and Bobrow, R.J. Semantic Interpretation Using KL-ONE. In *Proceedings of COLING-84 and the 22nd Annual Meeting of the Association for Computational Linguistics*, pages 101-107. Association for Computational Linguistics, Stanford, CA, July, 1984.

[33] Stallard, David. Answering Questions Posed in an Intensional Logic: A Multilevel Semantics Approach. In R. Weischedel, D.Ayuso, A. Haas, E. Hinrichs, R. Scha, V. Shaked, D. Stallard (editors), *Research and Development in Natural Language Understanding as Part of the Strategic Computing Program*, chapter 4, pages 35-47. BBN Laboratories, Cambridge, Mass., 1987. Report No. 6522.

[34] Vilain, M. The Restricted Language Architecture of a Hybrid Representation System. In *Proceedings of IJCAI85*, pages 547-551. International Joint Conferences on Artificial Intelligence, Inc., Morgan Kaufmann Publishers, Inc., Los Angeles, CA, August, 1985.

[35] Weischedel, R.M. Knowledge Representation and Natural Language Processing. *Proceedings of the IEEE* 74(7):905-920, July, 1986.

[36] Weischedel, R.M., Bobrow, R., Ayuso, D.M., and Ramshaw, L. Portability in the Janus Natural Language Interface. In *Speech and Natural Language*, pages 112-117. Morgan Kaufmann Publishers Inc., San Mateo, CA, 1989.

2. A View of Ill-Formed Input Processing

Ralph M. Weischedel[10]

Abstract

There are several purposes of this paper:

1. to define a class of phenomena that may be termed ill-formed input,
2. to differentiate this class from problems of spoken language understanding,
3. to briefly summarize the state of the art in understanding ill-formed input,
4. to indicate areas for further theoretical progress, and
5. to speculate regarding where there may be strong practical pay-off.

2.1 Introduction

Within the next ten years, natural language (NL) systems that process a broad range of ill-formed input are quite likely. Thus far, most research and development on ill-formedness has assumed that the goal is correct understanding, e.g., to carry out a user command or request to update a knowledge base given an incoming message, etc. This is highly demanding, since one must ascertain what a user intended at some level of meaning representation. In [41]; an intelligent tutor helped a (second) language student doing comprehension exercises; such a system must not only determine what the student meant but also diagnose language errors by the student. See [43] for evidence that diagnosing the violated well-formedness constraint(s) is critical even if the goal is not to correct error. Diagnosing the violated constraint(s) offers the potential of limiting the search space for an interpretation and of significant inferences based on the violation(s). For the remainder of the paper, we assume that the goal is understanding the input,[11] and that diagnosis of the difficulty in understanding is part of the goal.

In Section 2, definitions and examples of ill-formedness are provided. Section 2.3 reviews the state of the art. Section 2.4 describes the kind of predictive capability combining syntax and

[10] This paper originally appeared in the *Proceedings of the 1987 Natural Language Planning Workshop: Planning for Future Research Directions for the Next Decade*, 20-23 September 1987, Minnowbrook Conference Center, Blue Mountain Lake, NY.

[11] A goal not necessarily requiring correct understanding is stylistic analysis [10, 23], since a greater number of errors in flagging constructions for possible problems may be acceptable than where understanding or instruction is desired.

semantics offers, for instance, to predict a missing word or the meaning of an unknown word. In the three following sections, three future directions for work on ill-formedness are suggested. These are measuring the search space (Section 2.5), understanding some particularly challenging classes of problems (Section 2.6), and developing semantic-pragmatic interaction (Section 2.7). Our conclusions regarding potential payoff and future directions in the next five to ten years are in Section 2.8.

2.2 What Ill-formedness is

Consider the following definitions:

- An input is *absolutely ill-formed* if a native speaker would judge it to be ill-formed, e.g., something to be edited, an error, or not what was intended.
- An input is *relatively ill-formed* if there is no interpretation that would satisfy all of the NL ystem's well-formedness constraints, even though a native speaker may judge it well-formed.
- An input is *ill-formed* if it is either absolutely or relatively ill-formed.

In examples below, underlining is used to draw your attention to the source of the problem. Given the definitions above, the following kinds of problems are peculiar to written language:

- Typographical errors, e.g., *ohter*, instead of *other*. Typos may also result in recognizable words, as in *an* instead of *and*.
- Spelling errors, e.g., *Ralf* instead of *Ralph*.
- Punctuation errors, e.g., inserting or omitting commas incorrectly, misplacement or omission of apostrophes in possessives, etc.
- Homonym errors, e.g., *to* instead of *too*, or confusing *there*, *their*, and *they're*.

Each of the classes above result from human performance errors, and illustrate absolute ill-formedness.

Similarly, there are classes of ill-formedness peculiar to spoken language:

- Mispronunciations, e.g., saying that word as if it were spelled *mispronounciations*, or stressing the wrong syllable. Fromkin [14] has provided a taxonomy of human speech production errors that appear rule-based, as opposed to ungoverned or random occurrences.
- Spoonerisms, e.g., saying *fauter waucet* instead of *water faucet*.

Each of the classes above are human performance errors, resulting in absolute ill-formedness. However, the overwhelming variety of ill-formedness problems arise in both the spoken and written modality; examples of absolute ill-formedness include:

- Misreference, as in describing a purple object as *the blue one*.
- Word order switching as in saying *the terminal of the screen* when one meant *the screen of the terminal*.[12]
- Negation errors, e.g., *All doors will not open* when the train conductor meant *Not all doors will open*.
- Omitting words, as in *Send file printer* rather than the full form *Send the file to the printer*.[13]
- Subject-verb disagreement, as in *A particularly important and challenging collection of problems are relatively ill-formed and arise in both spoken and written language* or in *One of the overwhelming number of troubles that befell them are . . .*
- Resumptive pronouns and resumptive noun phrases, as in *The people that he told them about it*, where *them* is intended to be coreferential with *people*.
- Run together sentences, as if the person forgot how the sentence was started. An example collected from a written corpus [22] is *She couldn't expect to get a high standard salary and plus being so young*.
- Retarted sentences, as in *Some people many try to improve society*, which was also collected in a written corpus [22].
- Pronominal case errors, as in *between you and I*.
- Word order errors, as non-native speakers can make, e.g., *I wonder where is the problem*.

Some particularly important and challenging problems are relatively ill-formed and arise in both spoken and written language.

- Words unknown to the hearer or reader, but part of the language.
- Novel or unknown word senses, though the word itself is known. For instance, Navy jargon includes phrases such as *What is Stark's readiness?* Though that sublanguage does not include *preparedness* as a synonym for *readiness*, it would be useful for a system to be able to infer what a user means by the input *What is Stark's preparedness?*
- Novel (non-frozen) figures of speech, e.g., metaphor, metonymy, and synecdoche.
- Novel nominal compounds, as in *window aisle seat*, which was used by a stewardess on a wide-body jet.
- Violated presuppositions, as in *Did John fail to go?* when John did not try to go.

[12] Fromkin [14] has recorded these errors.

[13] Though this may seem to occur only in typed language this author has heard such omissions in spoken language Further, consider how many times when struggling for the appropriate word, you start the utterance over or someone supplies an appropriate word for you.

The lists above are not intended to be exhaustive. More thorough taxonomies of ill-formedness appear in [14, 22]. Statistical studies of frequency of occurrence for various classes of ill-formedness have been conducted for written data base access [11, 37]; those studies suggest that as much as 25-30% of typed input may be absolutely or relatively ill-formed.

From the definitions and examples, it is clear that

- Ill-formed input need not be ungrammatical; there may be no interpretation due to semantic or pragmatic problems.
- The NL system may not know *whether the input contains an error or whether its* models are too limited to process the input.
- Since there is no interpretation for the input, then one or more of the constraints of the NL system are violated; understanding ill-formed input therefore is a constraint satisfaction problem.
- Since one or more of the constraints are violated, relaxing constraints in order to find an interpretation will mean opening up the search space for an interpretation substantially.

2.2.1 Ill-formedness and Spoken Language

In earlier work, spoken language and ill-formed input seemed highly correlated due to the processing approach, even though they are not equivalent. For instance, in [13] a commercially available speech box provided the sequence of recognized words to a NL processor. Words incorrectly "heard" would have to be replaced by the natural language component; missing words would have to be hallucinated by the NL system; any extraneous words would have to be ignored. While the NL system employed ill-formedness techniques, a speech box that provides more than one alternative may have proposed the correct sequence, along with incorrect ones. Thus, ill-formedness processing was necessitated by the approach. In other work, [27, 39, 47], speech components provided several alternatives at each location where a word might occur; NL well-formedness constraints from syntax and semantics were used to eliminate possible word proposals, but the sequence finally selected was well-formed.

However, ill-formedness should not be deemed equivalent with the peculiar problems of mechanical recognition of continuous speech input. This is because ill-formedness occurs in both spoken and written modes. Therefore, even with perfect speech signal processing, spoken language systems must be prepared to handle ill-formed input. Human processing seems to allow for

- employing natural language constraints so predictively that we can often complete an incomplete utterance or hear what we expect (rather than what is actually uttered) and

- recognizing what is uttered, even if it is ill-formed (e.g., spoonerisms), yet understand what is meant by the speaker.

2.2.2 Limitations of Some Suggestions

In the last ten years, the following suggestions have frequently been expressed. While each contains useful elements, the common thread is proposing a partial solution which is too weak without further augmenting them to have a more complete account. Section 2.3 contains numerous references to more complete approaches.

2.2.2.1 Ignoring Constraints

Suppose the solution to ill-formed input is to remove oft-violated syntactic constraints. For instance, for agreement errors, encoding both the grammatical and ungrammatical forms could be proposed as the solution to understanding ill-formed input. However, ignoring subject-verb agreement in such a way would mean that the following sentence would be ambiguous: *List the frigates of the battle group that were deployed in the Persian Gulf.* Ignoring determiner-noun agreement can lead to quite bizarre parses, such as interpreting *one end called the top* as a pronoun *one* followed by an appositive *end called the top.* Instead, interpretations where constraints are completely satisfied should be ranked ahead of others, both to prefer interpretations and to trim the search space. Furthermore, ignoring constraints seems implausible for productive errors, such as spoonerisms, typos, and misreferences. It seems undesirable to load the dictionary with all possible spoonerisms and all possible rule-based mispronunciations identified in Fromkin's taxonomy [14], when compared to having a set of rules that predict those classes of ill-formedness.

2.2.2.2 Bottom-up Parsing

Suppose bottom-up parsing were the solution to understanding ill-formed input, since it provides a complete phrasal description of everything found in the input. Bottom-up parsing may be an element of a solution, but there are other critical issues to be addressed. Bottom-up parsing provides a forest of phrase structures; what is necessary in addition is a heuristic to determine which of the phrase structures are part of the intended message and how the intended phrases may be combined to form the intended meaning. Furthermore, proposing bottom-up parsing misses the reality that the ill-formedness may be semantic or pragmatic, rather than syntactic, in nature.

2.2.2.3 Semantics First

One could argue that applying semantic constraints first and using syntactic constraints purely for disambiguation is the correct approach. However, semantic constraints may themselves be violated. For instance, metonymy, metaphor, and synecdoche are cases where case frame constraints are deliberately violated; also, the case frame constraints may be incomplete or have too stringent rules. Even after adding semantic relaxation to the approach, one must further add techniques for dealing with *ungrammaticality* (for instance, when syntactic analysis is called to eliminate ambiguity but the ungrammaticality arises in exactly the analysis that is needed) and for handing pragmatic ill-formedness.

2.3 Prospects in Handling Ill-Formed Input

The overwhelming majority of researchers [7, 9, 12, 16, 17, 20, 33, 34, 35, 40, 43, 46] have investigated strategies employing both syntactic and semantic constraints to deal with a broad class of syntactic and/or semantic problems. All involve preference for well-formed interpretations and a means of allowing for certain constraints to be violated. Though the strategies proposed are highly varied and cover many classes of ill-formedness, there still seems to be a clear distinction between those classes of problems for which reasonably good syntactic and semantic strategies exist, and classes of ill-formedness that seem particularly intractable without a strong model of pragmatic knowledge for proper understanding. We present justifications for these claims in this section.

Strategies for the following classes of ill-formedness have been suggested using only syntactic and semantic knowledge:

- failed grammatical tests, e.g., subject-verb agreement [25, 30, 43],
- word confusions, e.g., homonyms, *good* for *well*, etc. [30, 43],
- typographical errors leading to unknown words [20],
- resumptive pronouns and resumptive noun phrases [43],
- selection restriction violation [16, 17, 43], and
- metaphor [12, 43].

Furthermore, unifying strategies for treating such problems have been suggested within various paradigms [7, 35, 43]. Nevertheless, a significant class of problems seems beyond reliance on purely syntactic and semantic grounds [44], and no unifying strategy has yet been worked out for them.

2.3.1 Elementary classes

Some classes of ill-formedness are quite easy to handle by almost any technique; examples include subject-verb agreement, determiner-noun agreement, errors in pronominal case, resumptive pronouns, and typographical errors resulting in an unknown word. For instance, a simple scan can identify a word the NL system does not recognize.

2.3.2 Challenging Classes

An experiment by Razi [33], helped to identify challenging classes of ill-formedness that cannot be handled by combining only morphological, syntactic, and semantic knowledge without pragmatics. The experiment suggested two results:

1. If only syntactic and semantic criteria (i.e., selection restrictions or case frame constraints) are available to guide the search for an interpretation, then the oft employed engineering heuristic [20, 23, 41] of exploring blocked interpretations ordered by the amount of input covered seems highly effective for ordering the search space. However, there are several factors which enhanced the heuristic's performance, such as lack of figurative language (e.g., metonymy and metaphor), lack of creative use of nominal compounding rules, absence of appositives, limited use of conjunction, relatively few case frames for verbs (so that case frames are highly constraining), and restriction to simple ellipsis structures (e.g., noun phrases and prepositional phrases only). Furthermore, the heuristic performs well only for simple problems, e.g., subject-verb disagreement, determiner-number errors, errors in pronominal case, resumptive pronouns, and typographical errors that lead to an unknown word, etc.
2. The work showed the limits of the approach and the limitations of employing only syntactic and semantic knowledge. In particular, for spelling/typographical errors producing a known word, for errors in possessive formation, for clauses run together, and for others discussed in the remainder of this section, the heuristic helps little even with the limiting factors mentioned in the previous paragraph. Furthermore, if those limiting factors are not in effect, the heuristic would be potentially far less effective even for simpler kinds of errors, such as subject-verb agreement.

Though Razi's experiment used a left-to-right, top-down, ATN parser, a brief follow-up consideration of the situation in a bottom-up context showed the same kinds of problems and limitations.

We conclude that those limitations will apply to any approach that uses only syntactic and semantic knowledge, rather than a model of the user's intent in making the request. In the remainder of this section we discuss a few classes of problems that seem beyond reliance on purely syntactic and semantic grounds [44]; no unifying strategy has yet been worked out for them.

2.3.2.1 Alias errors

For spelling/typographical errors that result in a known word, the system must mistrust its input, reading for at least one of the words present, some other word. The problem is combinatorially quite complex, since there is no overt evidence of which is the misspelled/mistyped word; the system merely knows that the input cannot be interpreted. At best, the system has a set of partial semantic representations regarding the meaning of the input, but no way to combine them into a complete thought.

Alias errors have been discussed in the literature [7, 44], but it appears that the only extensive implementation is that of Trawick [38]. He handled such errors by generating all likely respellings of all words in the input, which would be computationally tractable only in the most limited domains. No one has employed a model of user intention to determine what the input should have been.

2.3.2.2 Run-together sentences

By *run-together sentences*, we mean input that starts one way, but finishes in a different way, as if in a plan of what to say, the user/speaker forgot what the utterance plan was, or lost their place in the plan, etc. Like alias errors, the problem is that one has a set of partial semantic representations, none of which can be combined into a complete thought, unless one ignores large fragments of the input wholesale.

The general problem of run-together sentences has not been attempted to date, though a simpler case has been, namely, *restarted sentences*, where the first part may be ignored since the person totally restarted the utterance/input. In [19], a heuristic has been proposed for the simpler problem of restarted sentences, using only syntactic and semantic knowledge; it involves ignoring the first part of the input until the portion remaining on the right syntactically and semantically expresses a complete thought. Phonetic signals in speech input may suggest when an utterance has been restarted [21], but no speech recognition system has been developed yet that reliably recognizes such signals. Furthermore, no strategy for the more general problem ot run-together sentences has been proposed.

2.3.2.3 Pragmatic overshoot

Input exhibiting *pragmatic overshoot* is syntactically and semantically acceptable, but does not make sense pragmatically. For instance, the structural configuration of the data base may preclude data corresponding to the request; integrity constraints (or other axioms) on the data base content may be violated in the input. If the application is an expert system, it may have no functional capability corresponding to the user request.

The only research on this problem to date has been work by Carberry [5]. In that analysis the semantic interpretation of a request is examined to find subformulas, and substitutions for them, that would result in a semantic interpretation that makes sense pragmatically. At the core of the processing is an algorithm for building links between the semantic representation of the input and a model of the goals and plans of the user.

As an example, *What apartments are for sale?* cannot directly be interpreted by a data base that includes data on both **condominiums for** sale, **apartments for rent**, and **apartment buildings for sale**. The algorithm examines the (uninterpretable) semantic representation for *apartments for sale*, and finds that three substitutions into that formula make sense (yielding semantic representations corresponding to *condominiums for sale*, *apartments for rent*, and *apartment houses for sale*). The dialogue context is searched to see which, if any, is part of the system's model of the user's goals.

2.3.2.4 Contextual ellipsis

Contextual ellipsis is the use of fragments to convey a complete thought in context. [14] Though a number of heuristics have been developed in the last few years for contextual ellipsis, most [8, 20, 25, 42] are based on predicting the omitted (elided) material based on syntactic or semantic similarity with the previous input, using only minimal models of pragmatic context. In [4, 29], techniques are suggested for determining how a fragmentary semantic representation (corresponding to the interpretation of a noun phrase or prepositional phrase) could fit within the (complete) formulas that represent the system's model of the user's plans, domain goals, and discourse goals.

For instance, the last form in the dialogue below is elliptical.

User: *What job did Joe Brown interview for?*
System: *He'd like to be a control lab chemist.*
User: *Anne's evaluation?*

The semantic representation of the fragment *Anne's evaluation* can be linked to a goal for collecting evaluations of a job applicant, which fits well in a plan for making hiring decisions.

[14] Certainly one can argue that contextual ellipsis is well-formed, nevertheless many have treated it as relatively ill-formed and developed algorithms to infer the omitted portions of meaning.

2.3.2.5 Unknown words

Inferring the meaning of unknown words from context has been studied for some time. In [20], a technique is provided for inferring all that one can from syntactic and semantic knowledge. In [16], techniques were presented that could deduce the meaning of the unknown verb *scudded* in *Enemy scudded bombs at us* by using semantic knowledge about enemies and bombs. However, syntactic and semantic clues alone are often unable to resolve cases that pragmatics can handle. For instance, consider natural language access to computer mail systems [18, 24]. Entities in that domain are limited to messages, addresses, persons, sites, times, etc. Therefore, it is a very limited domain. Yet, if one says, *FROB the message to Jones*, almost any operation in the mail domain is a possibility, since *to Jones* could be dative or *the message to Jones* could be a reference. If both interpretations for *to Jones* are possible, then *answer, send, resend, forward, delete, move*, etc. are all possible on syntactic and semantic grounds.

Additional knowledge of typical, general goals of an agent [6] and of goals specific to situations representative of prototypical scenarios [15, 16] have been effectively employed to infer the meaning of nouns and verbs. The partial semantic representation in this case might be simply a logical form with a variable where the semantics of the unknown word should appear.

2.3.3 Ranking alternative interpretations

With an ill-formed input, an unsatisfied constraint preventing an interpretation from being found could arise from problems in the input or alternatively from deficiencies in the understanding system, as indicated in the table below. A natural language processing system has no foolproof way of knowing whether the problem is with the user's input or with its own limited model of language.

Input Error	System Problem
an error in an input symbol	inadequate lexical information
ungrammaticality	inadequate grammar
a semantic error	incomplete selection restrictions
	overly restrictive case frame constraints
	a figure of speech
non-felicitous input	incomplete dialog models.

In the face of all the alternatives for what might prevent the system from understanding the input, all the knowledge and constraints available must be applied to determine what is intended.

Using goals and plans of the user to select among competing (potentially partial) interpretations of an ill-formed input has not previously been studied. Only measures based on syntactic and semantic criteria have been proposed thus far [9, 31]. We postulate that the kinds of heuristics described in Section 2.6 for ranking partial semantic interpretations regarding which fit best in the context of the user's plans and goals will also rank interpretations resulting from alternative hypotheses regarding the diagnosis and correction of an ill-formed input.

2.4 Exploring Syntactic and Semantic Predictions

In this section, we explore what syntactic and semantic constraints can offer in predicting what words can come next. This can be invaluable in dealing with unknown words. Also, as will become clearer in the next section, it may offer much in continuous speech recognition. We will try to make minimal assumptions, though it is impossible not to make some (and even to be unaware of others made implicitly). Suppose that words are first identified as to class, e.g., noun, verb, adjective, etc. The particular choice of categories, whether more fine-grained or more semantic than syntactic, is not important. To avoid making any assumption about whether processing is bottom up or top down, first consider a phrase independent of its context.

Suppose *C1* is an adjective whose selection restriction requires that it modify vessels. If we see a phrase, "... *the C1* ...", many syntactic categories could come next, including adverbs, adjectives, present or past participles of verbs, nouns, proper nouns, conjunctions, and prepositions. Thus, in this example, the syntactic context is hardly constraining. Second, there are semantic constraints depending on the type of the next word:

- If the next word were an adjective, then it too must have a selection restriction satisfiable by a vessel. For instance, *harpoon capable* would satisfy the constraint, but *sad* would not.
- If the next word were a present participle of a verb, there are two cases. An intransitive verb would be consistent with *the C1* if it constrains its logical subject to be consistent with a vessel. Therefore, *downgrade* (as in *Midway downgraded to C3 yesterday*) would be consistent, but *command* would not be. However, if the verb is transitive, then it should constrain its logical object to be consistent with a vessel; *command* would be consistent.
- If the next word were a past participle of a verb, then it should constrain its logical object to be consistent with a vessel. Thus, *deploy* would be consistent.
- If the next word were a noun, there are again two cases. If it serves as the head noun, it must be consistent with being a vessel. Thus, *unit* and *carrier* are possible, but *admiral* is not. On the other hand, the noun could be the beginning of a nominal compound; if any predictions are to be made, one would need a means of predicting all possible beginnings of nominal compounds that describe a vessel, a potentially insurmountable task.
- If the next word were a proper noun, it must be consistent with being a vessel. Thus, *Enterprise* is consistent, but *Pearl Harbor* is not.

- If the next word were a conjunction, it could be a coordinating conjunction, e.g., *and* or *or*, or a subordinating conjunction, which could arise in constructions like *the unemployed while suffering much are* ...
- If the next word were a preposition, it must be able to relate entities consistent with type vessel to the type of the object of the preposition. Thus, *in, from, of*, etc. are all consistent, but *during* is not.
- If the next word were an adverb, no constraining semantic check seems available, since it most likely modifies an adjective or adverb one word further left.

Obviously, combining syntactic and semantic constraints should provide more predictive power than just applying one of the two sources of knowledge. Unfortunately, no data is available to indicate as yet how much each knowledge source contributes, nor how much they together contribute. Such empirical analysis is highly desirable. A suggestion regarding an appropriate measure appears in the next section.

There are several things to note about the processing above:

- We have assumed that syntactic and semanuc information can be brought to bear on a word by word basis to maximize the predictive power available from syntactic and semantic constraints.
- No particular formalism is assumed. Whatever the formalism, one must be able to predict what word categories or literal words may come next given a partially recognized string corresponding to a partially matched constituent.
- Naturally, though the example assumed we needed predictions about a word to the right, similar analysis could be given for what words could appear to the left of a processed word sequence.
- Were top-down information available to constrain the context in which a phrase occurs, the predictive power should be greater, in principle. There does not as yet seem to be any clear data to indicate how much that would further constrain the possibilities. Furthermore, any gain in semantic predictive power is diminished by the fact that whole noun phrases can appear before the head noun is encountered. For instance, suppose the left context is *Enterprise will be commanded by* .. ; though a person is required, the noun phrase can start as *the C1 cruiser's commanding officer*. Though *the C1* ... would predict a vessel specification, and *commanded by* predicts a person, there is nothing inconsistent with continuing as shown.
- Non-frozen figures of speech are an exception to the processing above, since they will in general appear to violate semantic predictions. In Section 2.7, we discuss ways one could approach this problem.

This predictive power of syntactic and semantic context is potentially relevant to certain classes of ill-formed input. If one encounters an unknown word, the combination of syntactic and semantic predictions from the left and right contexts provide some information regarding syntax constraints and selection restrictions of the word. Several efforts [2, 6, 15, 16, 20] have investigated this.

Also, alias errors could be treated like unknown words. The NL processor could process a string of n words n times, each time as if one of the words were unknown to find out the syntactic and semantic constraints that apply to that word were it the one mistyped/misspelled. Knowledge of typical spelling errors and typical typographical errors could further be applied to determine what the user may have meant. For further detail about this proposed approach, see [32].

The novel aspect of what we have proposed above is the ability to employ semantic constraints before finding the head word of the phrase, e.g., the main verb of a clause or the main noun of a noun phrase, and to infer information about a word other than the head. It appears that much previous work [2, 6, 15, 16] employed syntactic and semantic predictions, e.g., to infer the meaning of an unknown head word only.

2.5 A Possible Measure for the Contribution of Syntactic and Semantic Knowledge

In the previous section, we discussed limitations of syntactic and semantic knowledge in the context of understanding an ill-formed input. An open issue in evaluating NL theories and systems is measuring their effectiveness; in this context, an important question is measuring the contribution of syntactic and semantic knowledge in trimming the search space of an NL processor when attempting to understand an ill-formed input.

One possibility for such measurement may be *perplexity*, which has proved valuable in speech recognition. For further information on the following definitions, see [3]. One can define *entropy* for a set of m events as follows:

$$H = -\sum_{i=1}^{m} p_i \log_2 p_i,$$

where p_i is the probability of event i occurring. The *perplexity* is:

$$Q = 2^H.$$

Now consider the utterance of a sequence of words $w_1 \ldots w_n$ as an event, and assume that some statistical distribution can be associated with a language L. The average entropy per word may be defined as

$$h = \lim_{n \to \infty} -\frac{1}{n} E\{\log_2 p(w_1 \ldots w_n)\},$$

where $E\{\ \}$ is the expected value. One can estimate the perplexity Q for a given corpus consisting of a word sequence of length n, $w_1 \ldots w_n$ in the following way:

$$T = -\frac{1}{n} \log_2 p(w_1 \ldots w_n) \qquad Q = 2^T.$$

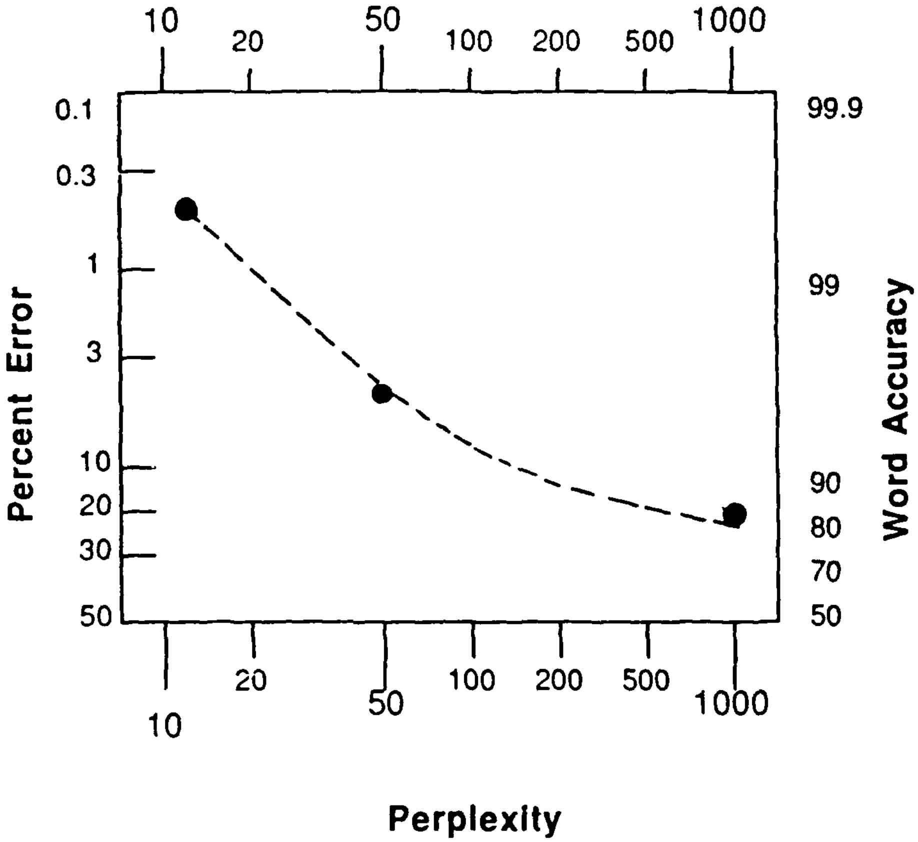

Figure 2-1: Error Rate as a Function of Perplexity

The importance of perplexity for speech recognition can be seen in the graph in 2-1.The results are from an experiment conducted by the BBN speech recognition group. Each of the scales is logarithmic. Across the bottom (and the top) is increasing perplexity. The left scale is the percentage of errors in predicting words given a test set of continuous spoken language input; note that the scale is printed with lowest percentage error at the top. The right scale is the percent correct word recognition, i.e., one hundred minus the error rate.

Note that with increasing perplexity, the error rate increases, and the percentage of correct word recognition decreases. This suggests that the goal of applying syntactic, semantic, and pragmatic knowledge to spoken language processing is the effective reduction of perplexity. Since there are ways of estimating the perplexity of a language, one can attempt to measure the effectiveness of alternative NL architectures, heuristics, etc., in contributing to acoustic-phonetic knowledge.

We speculate that perplexity might be a useful measure regarding certain issues in ill-formedness processing. Since processing ill-formedness implies relaxing well-formedness constraints that could constrain search were ill-formedness ignored, the size of the search space is a crucial issue in evaluating a system's ability to understand an ill-formed input. For instance, what is the size of the search space for well-formed language? How much does the search space grow given the addition of classes of ill-formedness? Given alternative systems, architectures, heuristics, etc. that cover comparable subsets of language, does one design alternative reduce search with significantly greater effectiveness? How effective is syntactic knowledge in constraining search? How effective is semantic knowledge in constraining search? How does percentage of correct understanding, runtime performance, or other measures vary as the search space grows? Since perplexity offers a way of measuring the effective search that a system encounters, it may be fruitful to investigate using it to answer these and other questions.

2.6 An Approach to the Challenging Classes

We have argued in Section 2.3 and elsewhere [44] that syntactic and semantic knowledge must be augmented by pragmatic knowledge in order to understand the difficult classes of ill-formedness including alias errors, run-together sentences, pragmatic overshoot, contextual ellipsis, and unknown words. We further argued that determining what an ill-formed input means is also a largely unsolved problem in ranking alternatives. The hypothesis we propose has two parts:

1. These phenomena can be handled by a common core of pragmatic processing to be added to existing syntactic and semantic techniques. The new processing to be added to the state of the art involves building links between the fragmentary formulas representing the semantic representation of the input and a model of the goals and plans of the user.
2. That same pragmatic interpretation process is critical to determining whether a semantic representation of a well-formed input is coherent in dialogue context.

Work on pragmatic overshoot [4], on contextual ellipsis [4], and on unknown words [6, 15, 16] all support this hypothesis. In this section, we discuss ongoing work by Ramshaw [32] to provide additional support, and to illustrate how a process might work within this approach. Our focus is knowledge about the plans and goals of a speaker in the framework first proposed by Allen [1] and followed up by others [4, 28, 36]. Therefore, we can assume we have available a tree representing the goals and subgoals that a user may wish to accomplish. A particular path in that tree represents the stack of pending goals that the user may have.

Consider the following example of an alias error: *I want a copy of Robinson Crusoe. Is Wordsworth Books open in Sunday?* Ramshaw [32] has proposed that a heuristic for alias errors is to try for interpretations where each word in turn is taken as a wildcard, WILD. For this example, that would be six cases. The one where *in* is replaced with *WILD* is *Is Wordsworth Books open WILD Sunday?* Suppose a semantic interpretation for that is

QUERY[open(WB t) & WILD'(t S),]

where WB is the constant corresponding to *Wordsworth Books* and S is the constant corresponding to the next Sunday. The planning model at the time of the ill-formedness *Is Wordsworth Books open in Sunday?* should have as its topmost goal, **possess(speaker, x)**, where x corresponds to *a copy of Robinson Crusoe*. Since possessing something can be achieved by purchasing it, the next most specific goal could be **purchase(speaker, x)**. However, a precondition of that goal is that the store be open during the time of the purchase. Therefore, the predicate **WILD'** *might* be **during**, which can be lexicalized as *on* a day. Furthermore, typing *in* for *on* is quite believable. Thus, finding a link between the formula

QUERY[open(WB t) & WILD'(t S),]

and the planning context could suggest what the user meant.

The analysis there might at first seem like overkill; however, there are several supporting reasons for developing such techniques. First, there is no overt evidence as to which word should be corrected. Furthermore, the only alternative strategy suggested thus far [38] appears to be exponential in the number of words in the sentence, attempting interpretations for all combinations of typographically close words for every word in the sentence. Second, the system has no way of knowing that the problem is an alias error. Perhaps one of the noun phrases (e.g., *Sunday*) is being used metonymically; perhaps there is a sense of *open in* that it does not know (and should learn); perhaps it is two thoughts run together; etc. Third, the model of plan context in independently justifiable in the NL system. It is well known that cooperative question answering must take user goals and plans into account in order to correctly address the user's needs, answer at an appropriate level of detail, identify unfulfillable goals, and suggest alternative plans for attaining the user's goals. Achieving these four qualities requires identifying what the user meant by a request (whether well-formed or not), and how that fits in with his/her current plans and goals. Fourth, the linking described in the example above might be common to processing alias errors such as above, to making sense of run-together sentences, to interpreting pragmatic overshoot, to understanding contextual ellipsis, to interpreting unknown words, to ranking alternative interpretations of an ill-formed input, and to recognizing coherence in dialogue.

2.7 Semantic - Pragmatic Interaction

One form in which we can bring pragmatics to bear is described in the previous section. Namely, one can send pragmatics well-formed formulas (possibly containing a wildcard where a constant of the logic should appear) for pragmatics to process, for instance, to constrain what constants could appear for a wildcard. However, a second kind of communication between semantics and pragmatics is interaction. In this section, we argue that this is highly desirable, illustrating the point with the problem of understanding metonymy.[15]

Non-frozen figures of speech, though creative, do seem to fall into regular patterns. Many of the examples below are from a taxonomy for metonymy by Lakoff [26]; the following rules[16] might be used to accept a description which violates selectional restrictions unless interpreted as a figure of speech

- Description A of semantic type A' may be accepted, when a type B' is required, if A describes a part of some whole B which has type B'. Then assume B is designated by A. An example is *How many hands were on Stark?*
- Description A of semantic type A' may be accepted, when a type B' is required, if A produces some product B of type B'. Assume B is designated by A. Consider the example *Is there an Apple onboard?*
- Description A of semantic type A' may be accepted, when a type B' is required, if A is an object used by some B of type B'. Assume B is designated by A. An example is *The trumpet has a cold sore.*
- Description A of semantic type <person> may be accepted, when a type B' is required, if A denotes a person who controls some B of type B'. Assume B is designated by A. An example is *Did the President shell Iran's coast?*
- Description A of semantic type A' may be accepted, when a person is required, if A is an institution directed by some people B. Assume B is designated by A. (One could alternatively argue that one's type hierarchy should include a type <legal-person> which includes both <person> and <institution>.) An example is *Enterprise said it is low on fuel.*
- Description A of semantic type A' may be accepted, when a type B' is required, if A denotes a location of some institution B of type B'. Assume B is designated by A. Consider the example *Pearl Harbor ordered New Jersey to the Persian Gulf.*
- Description A of semantic type A' may be accepted, when a type B' is required, if A denotes a location of some well-known event of type B'. Assume B is designated by A. Consider the example *Pearl Harbor caused the USA to enter World War II.*
- Description A of semantic type <display-object> may be accepted, when a type B' is required, if A denotes an image representing some object B of type B'. Assume B is designated by A. Consider the example *Send the blue triangle 300 miles NE.*
- Description A of semantic type A' may be accepted, when a type <display-object> is required, if A is an object that can be represented by some display entity B of type <display-object>. Assume B is designated by A. Consider the example *Show Stark*, where one means *Show an icon representing Stark.*

[15] Whether one treats creative cases of metonymy as well-formed or relatively ill-formed does not matter for the purposes of this discussion.

[16] An earlier version of these rules was reported in [45].

Note that each of these assumes detailed, specific pragmatic knowledge, e.g., typical subpart relationships, what is typically produced by a well-known institution, persons typically associated with a given object and its use, typical entities controlled by a given individual, the location of a given institution, and the location of a well-known event. While generic evidence may be sufficient, many examples require explicit knowledge about the particular entities described, not just their class or type, to understand the metonymy. For instance, *Pearl Harbor caused us to enter World War II* seems to require knowledge about a concrete well-known event associated with Pearl Harbor, if it is to be understood in any other than a shallow way. Similarly, *10 Downing Street announced new austerity measures today* seems to assume explicit common knowledge about what is located at 10 Downing Street, not just information about classes and types, if it is to be understood. The fact that pragmatic information is critical, rather than only semantic class information, strongly suggests the need for effective communication between semantics and pragmatics. Otherwise, NL systems would potentially have to accept any description metonymically with no effective control over what descriptions can reasonably substitute metonymically for another.

We believe that semantic-pragmatic interaction is not only a promising area for research but also may have great impact in the functionality of NL understanding systems.

2.8 Conclusions

2.8.1 Areas of potential pay-off

Processing ill-formed input seems critical in natural language interfaces to application systems, e.g., data bases, expert systems, decision support, etc., due to its frequency of occurrence [11, 37]. Furthermore, certain classes of messages may contain a substantial percentage of forms that are ill-formed. Spoken language contains a diverse class of ill-formedness, such as run-together sentences and mispronunciations [14]. Though it is not the case that every input or utterance need be understood automatically, the fact that a substantial percentage is required in each application above demands a model of understanding as much as possible and engaging in clarification dialogue otherwise. We believe the state of the art in understanding ill-formedness is already sufficiently advanced that NL interfaces, NL message processors, and new architectures for NL systems should include handling of subject-verb agreement errors, determiner-number agreement errors, incorrect pronominal case, resumptive pronouns, omitted determiners, omitted prepositions, and typographical errors leading to an unknown word.

In addition to incorporating the added functionality listed above in NL systems, one can envision applications based around ill-formedness processing such as authoring aids that hypothesize when a construction may be incorrect or applications such as computer-assisted language instruction tools that identify classes of student errors and that gear instruction accordingly. Given that work has been begun related to both applications [23, 41] commercial products are clearly technically feasible in the next five years.

2.8.2 Theoretical directions

Our belief is that the ability of NL systems to understand ill-formed input is limited by their inability to employ pragmatic information, such as a model of the plans and goals that a user has in mind, or factual knowledge that would enable reliable understanding of creative metonymic expressions. Furthermore, the most advanced R&D systems may reach a plateau in their ability to understand ill-formedness during the next five years, if they have not already reached it. To get off of that plateau, the following four areas for theoretical work are suggested based on their potential for massively improving the state of the art within five to ten years.

1. Developing effective interaction between semantic knowledge and pragmatic knowledge to have further evidence regarding what the person may have meant.
2. Discovering a common core strategy for employing a model of user plans and goals to determine what the user meant, particularly for problems such as words used in an unknown sense, alias errors, pragmatic overshoot, ellipsis, and run-together sentences.
3. Empirically measuring the effectiveness of the contribution of syntactic knowledge and of semantic knowledge to constrain the search space when trying to understand an ill-formed input.
4. Developing an effective strategy for clarification dialogue for the cases where no single interpretation is judged to be what the user may have meant.

This paper has focused on the first three of those topics.

Acknowledgements

Many thanks to Salim Roucos for his discussions of perplexity and the data reported in Section 2.5.

References

[1] Allen, J.F. and Perrault, C.R. Analyzing Intention in Utterances. *Artificial Intelligence* 15(3), December, 1980.

[2] Ayuso, D.M., Shaked, V., and Weischedel, R.M. An Environment for Acquiring Semantic Information. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pages 32-40. ACL, 1987.

[3] Bahl, L.R., Jelinek, F. and Mercer, R.L. A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-5(2):179-190, March, 1983.

[4] Carberry, M.S. A Pragmatics-Based Approach to Understanding Intersentential Ellipsis. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, pages 188-197. Association for Computational Linguistics, Chicago, IL, July, 1985.

[5] Carberry, M.S. Using Inferred Knowledge to Understand Pragmatically Ill-Formed Queries. *Communication Failure in Dialogue*. North-Holland, 1987.

[6] Carbonell, J.G. Towards a Self-Extending Parser. In *17th Annual Meeting of the Association for Computational Linguistics*, pages 3-7. Association for Computational Linguistics, La Jolla, CA, August, 1979.

[7] Carbonell, J.G. and Hayes, P.J. Recovery Strategies for Parsing Extragrammatical Language. *American Journal of Computational Linguistics* 9(3-4):123-146, 1983.

[8] Carbonell, J.G. Disourse Pragmatics and Ellipsis Resolution in Task-Oriented Natural Language Interfaces. In *Proceedings of the 21st Annual Meeting of the Association for Compuational Linguistics*, pages 164-168. Association for Computational Linguistics, June, 1983.

[9] Charniak, E. A Common Representation for Problem-Solving and Language Comprehension Information. *Artificial Intelligence* 12:225-255, 1981.

[10] Cherry, L.L. and Vesterman, W. *Writing Tools - The STYLE and DICTION Programs*. Technical Report 9, Computing Science, Bell Laboratories, Murray Hill, NJ, 1980.

[11] Eastman, C.M. and McLean, D.S. On the Need for Parsing Ill-Formed Input. *American Journal of Computational Linguistics* 7(4):257, October-December, 1981.

[12] Fass, D. and Wilks, Y. Preference Semantics, Ill-Formedness, and Metaphor. *American Journal of Computational Linguistics* 9(3-4):178-187, 1983.

[13] Fink, P.E., Biermann, A.W. The Correction of Ill-Formed Input Using History-Based Expectation with Applications to Speech Understanding. *Computational Linguistics* 12(1):13-36, January-March, 1986.

[14] Fromkin, V.A. *Janua Linguarum, Series maior 77: Speech Errors as Linguistic Evidence*. Mouton, The Hague, 1973.

[15] Granger Jr., R.H. Foul-Up: A program that figures out meanings of words from context. In *5th International Joint Conference on Artificial Intelligence - 1977.* IJCAI, 1977.

[16] Granger, R.H. The NOMAD System: Expectation-Based Detection and Correction of Errors during Understanding of Syntactically and Semantically Ill-Formed Text. *American Journal of Computational Linguistics* 9(3-4):188-198, 1983.

[17] Grishman, R., Hirschman, L., and Nhan, N.T. Discovery Procedures for Sublanguage Selectional Patterns: Initial Experiments. *Computational Linguistics* 12(3):205-215, July-September, 1986.

[18] Hayes, P.J. and Reddy, R. *An Anatomy of Graceful Interaction in Man-Machine Communcation.* Technical Report, Carnegie-Mellon University, 1979.

[19] Hayes, P. and Mouradian, G. Flexible Parsing. In *Proceedings of the 18th Annual Meeting of the Association for Computational Linguistics and Parasession on Topics on Interactive Discourse*, pages 97-103. Association for Computational Linguistics, Philadelphia, June, 1980.

[20] Hendrix, G., et al. Developing a Natural Language Interface to Complex Data. *ACM Transactions on Database Systems* 3(2):105-147, 1978.

[21] Hindle, D. Deterministic Parsing of Syntactic Non-fluencies. In *21st Annual Meeting of the Association for Computational Linguistics - Proceedings of the Conference.* ACL, 1983.

[22] Hull, G. and Bartholomae, D. *An Error Taxonomy.* Technical Report, Learning Research and Development Center, University of Pittsburgh, September, 1984.

[23] Jensen, K., Heidorn, G.E., Miller, L.A., and Ravin, Y. Parse Filling and Prose Fixing: Getting a Hold on Ill-Formedness. *American Journal of Computational Linguistics* 9(3-4):147-160, 1983.

[24] Kaczmarek, T., Mark, W., and Sondheimer, N. The Consul/CUE Interface: An Integrated Interactive Environment. In *Proceedings of CHI '83 Human Factors in Computing Systems*, pages 98-102 ACM, December, 1983.

[25] Kwasny, S.C. and Sondheimer, N.K. Relaxation Techniques for Parsing Grammatically Ill-formed Input in Natural Language Understanding Systems. *American Journal of Computational Linguistics* 7(2):99-108, 1981.

[26] Lakoff, G. and Johnson, M. *Metaphors We Live By.* The University of Chicago Press, Chicago, 1980.

[27] Lesser, V.R., Fennell, R.D., Erman, L.D., and Reddy, D.R. Organization of the Hearsay II Speech Understanding System. *IEEE Trans. Acoustics, Speech, and Signal Processing* ASSP-23"(1):11-24, 1975.

[28] Litman, D.J. and Allen, J.F. A Plan Recognition Model for Clarification Subdialogues. In *Proceedings of Coling84 and the 22nd Annual Meeting of the Association for Computational Linguistics*, pages 302-311. Association for Computational Linguistics, July, 1984.

[29] Litman, D.J. *Plan Recognition and Discourse Analysis: An Integrated Approach for Understanding Dialogue.* PhD thesis, U. of Rochester, 1985.

[30] Miller, L.A., Heidorn, G.E., and Jensen, K. Text-critiquing with the EPISTLE System: an Author's Aid to Better Syntax. In *AFIPS Conference Proceedings - 1981 NCC*. AFIPS Press, Montvale, NJ, 1981.

[31] Minton, S., Hayes, P.J. and Fain, J. Controlling Search in Flexible Parsing. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*. IJCAI, 1985.

[32] Ramshaw, L.A. *Pragmatic Knowledge for Resolving Ill-Formedness*. Technical Report Report 87-15, Department of Computer & Information Sciences, University of Delaware, 1987.

[33] Razi, A.M. *An Empirical Study of Robust Natural Language Processing*. PhD thesis, University of Delaware, June, 1985.

[34] Schank, R.C., Lebowitz, M., and Birnbaum, L. An Integrated Understander. *American Journal of Computational Linguistics* 6(1):13-30, 1980.

[35] Selfridge, M. Integrated Processing Produces Robust Understanding. *CL* 12(2), April-June, 1986.

[36] Sidner, C.L. Plan Parsing for Intended Response Recognition in Discourse. *Computational Intelligence* 1(1):1-10, February, 1985.

[37] Thompson, B.H. Linguistic Analysis of Natural Language Communication with Computers. In *Proceedings of the Eighth International Conference on Computational Linguistics*, pages 190-201. International Committee on Computational Linguistics, October, 1980.

[38] Trawick, D.J. *Robust Sentence Analysis and Habitability*. PhD thesis, California Institute of Technology, February, 1983.

[39] Walker, D. *Understanding Spoken Language*. Elsevier North-Holland, New York, New York, 1978.

[40] Waltz, D.L. An English Language Question Answering System for a Large Relational Database. *Communications of the ACM* 21(7):526-539, 1978.

[41] Weischedel, R.M., Voge, W., and James, M. An Artificial Intelligence Approach to Language Instruction. *Artificial Intelligence* 10:225-240, 1978.

[42] Weischedel, R.M. and Sondheimer, N.K. An Improved Heuristic for Ellipsis Processing. In *Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics*, pages 85-88. Association for Computational Linguistics, Jun, 1982.

[43] Weischedel, R. M. and Sondheimer, N. K. Meta-rules as a Basis for Processing Ill-Formed Input. *American Journal of Computational Linguistics* 9(3-4):161-177, 1983.

[44] Weischedel, R.M. and Ramshaw, L.A. Reflections on the Knowledge Needed to Process Ill-Formed Language. In S. Nirenburg (editor), *Machine Translation: Theoretical and Methodological Issues*. Cambridge University Press, Cambridge, England, 1987.

[45] Weischedel, R.M. and Sondheimer, N.K. *Relaxing Constraints in MIFIKL*. Technical Report, USC/Information Sciences Institute, 1983.

[46] Wilks, Y. Natural Language Understanding Systems Within the AI Paradigm - A Survey With Some Comparisons. *American Journal of Computational Linguistics* , 1976. Microfiche 40.

[47] Woods, W.A., Bates, M., Brown, G., Bruce, B., Cook, C., Klovstad, J., Makhoul, J., Nash-Webber, B., Schwartz, R., Wolf, J., Zue, V. *Speech Understanding Systems, Final Technical Progress Report, Volumes I-V.* Technical Report 3848, BBN, 1976.