Semantic Content Analysis: A New Methodology for The RELATUS Natural Language Environment

John C. Mallery
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
545 Technology Square, NE43-797
Cambridge, MA 02139-4301 USA
Phone: (617) 253-5966
Internet: JCMA@AI.MIT.EDU

in *Artificial Intelligence and International Politics,* editor, Valerie M. Hudson,Boulder: Westview Press, 1991.

Abstract

Semantic content analysis differs from traditional computerized content analysis because it operates on the referentially integrated, meaning representation of a text instead of a linear string of words. Rather than assessing the thematic orientation of texts based on the frequencies of word occurrences, this new methodology examines and interprets explicit knowledge representations of texts. There are three phases to a semantic content analysis:

- **Text Representation:** the sentences of a text are syntactically parsed and semantically represented to create meaning-rich text models;
- **Classification:** the political analyst applies recognizers, designed in advance, to classify relational configurations of words in text models;
- **Inspection:** the analyst uses any number of interfaces for inspecting text models to view the classifications.

In the RELATUS Natural Language Environment, lexical recognizers "tag" instances of categories by matching constraint descriptions for alternate lexical realizations (paraphrases in surface semantics). Word senses are semantically disambiguated by incorporating selection constraints into the descriptions that select correct lexical realizations. An advanced definition interface allows users to supply English sentences or noun phrases to specify constraint descriptions that retrieve and label lexical realizations. This frees the user from the need to know the details of the constraint language. Composite patterns can be detected by hierarchical application of lexical recognizers. Beyond semantic content analysis, lexical classification expands the referential performance because it provides a basic inference mechanism to extend indexation, semantically disambiguate word senses, and provide criteria for further deliberation in reference.

Contents

List of Figures

1 Introduction

Classification is the fundamental interpretive process found in political cognition, whether individual or collective. Recall that Ronald Reagan's "freedom fighters" and Leonid Brezhnev's "bandits and criminals" were two classifications for the same Afghanistan resistance. These leaders selected labels to call forth desired political actions in their audiences. The intended perlocutionary effects arise because the labels elicit particular ways of framing the problem, or "seeing-as" (Wittgenstein, 1953), and from these stereotypical, or "natural" political responses follow. Examples such as this suggest that classification is a fundamental political act (Mefford, 1988a) standing behind conflict and cooperation.

The basic cognitive process at work in classification involves an interplay between the micro-classifications of actions and relationships that call forth particular constructions of situations. These constructions, in turn, demand concordant micro classifications.[1] Paul Ricoeur (1971) has dubbed this the *double hermeneutic*. When problem framing is freed from temporal anchors in the present, this classificational process shades into analogical reasoning, or precedent logics, which has been proposed to model political understanding and organizational decision-making (Alker & Christensen, 1972; Alker, Bennet & Mefford, 1980; Mallery & Hurwitz, 1987; Mefford, 1987; Mallery, 1988a: 38-47). Because classification is a central cognitive process with high relevance for politics, tools to simulate classification are a high research priority. They can help reveal the cognitive dimensions of the phenomena as they yield methods for *computational politics* – the subfield of political science that uses symbolic reasoning techniques from artificial intelligence to formally model politics.

This paper introduces *semantic content analysis*, a methodology whose vehicle is automatic recognition and classification of instances in the knowledge representations of texts, or *text models*. Here, the RELATUS Natural Language System provides an environment that allows an analyst to syntactically parse and semantically represent a variety of English texts. A lexical (word) classification system that operates on text models allows the analyst to find micro-classifications, and in this way, perform a semantic content analysis of texts. Earlier computerized content-analytic techniques[2] could suggest the thematic orientations expressed in particular textual corpora, but they could neither reliably identify the sign (positive or negative mention) nor determine the *direction* of value or goal orientations. Instead of using word frequencies to indicate thematic orientation, semantic content analysis examines explicit structural relationships and inherent interpretations in text models.

---

[1] *Micro-classification* refers to fine-grained, bottom-up recognition of category instances whereas *macro-classification* suggests a course grained, top-down situation or expectation driven category recognition.

[2] Computerized content analysis deploys a form of keyword search to find the frequencies and correlations of interesting words. Word frequency is deemed to reflect salience, and in turn, importance for a speaker. Word senses are disambiguated according to word co-occurrences. But, standard content analysis programs neither analyze syntactic structures nor construct referentially-integrated semantic representations. Holsti (1968), Krippendorf (1980), and Weber (1985) provide overviews of content analysis. Duffy (1987, 1989) explains how computational hermeneutics transcends traditional content analysis.

Semantic content analysis differs from traditional content analysis because it operates on referentially-integrated text models. *Referential integration* means that references to the same object or relation, which may appear in different sentences of a text, are resolved and represented as the same semantic node. Text models are created incrementally through a process that syntactically analyzes each sentence (section 2.4.3), constructs a logical form (section 2.4.4). The reference system (section 2.4.2) uses the logical form to merge intersentential coreferences into referentially-integrated semantic representations (section 2.4.1). After syntactic analysis captures structural relations within sentences, referential integration compiles the information across sentences. Both are critically important for political texts because they make explicit *who does what to whom* throughout a text.

There are three phases to semantic content analysis:

- **Text Representation:** The sentences of a text are syntactically parsed and semantically represented to create referentially-integrated text models;
- **Classification:** The political analyst applies lexical recognizers, designed in advance, to locate and classify relational configurations in the text model;
- **Inspection:** The analyst uses any number of interfaces for inspecting the text model to view the classifications.

Lexical recognizers store constraint descriptions for different paraphrases, or alternate lexical realizations in surface semantics. Recognizers can incorporate *selection constraints* so their constraint descriptions will disambiguate words senses, picking out correct instances. Category instances are found as these constraint descriptions match knowledge structures in the text model. Lexical recognizers do not attach a tag word to sentence strings like conventional content analysis. Instead, each recognizer stores a separate constraint description that denotes the knowledge structure for the concept it recognizes. As recognizers locate instances, they label them with a subsumption relation from the knowledge structure for the concept to the instance knowledge structure. As lexical recognizers find and label instances, explicit taxonomic structures grow up from the instances toward more general and abstract categories.

An advanced definition interface (section 3.1.4) allows users to supply English sentences or noun phrases to specify constraint descriptions that retrieve lexical realizations and denote the associated concept. This frees the user from the need to know the details of the constraint language used by the RELATUS reference system. When a user wishes to recognize complex knowledge structures, the recognition task may be delegated to simpler recognizers whose annotations are combined when they are applied hierarchically. *Hierarchical lexical classification* (section 3.1.5) promotes clean abstractions by keeping individual recognizers simple – and simpler recognizers facilitate specification with English sentences.

Beyond semantic content analysis, lexical classification extends the referential capabilities of RELATUS because it provides a basic inference mechanism that augments indexation, semantically disambiguates word senses, and provides criteria for further deliberation in reference (section 3.2). As it helps extend the model reference (section 2.3), lexical classification in turn leads to more sophisticated text models that better approximate human linguistic and cognitive capabilities.

1.1 Relevance of Text Modeling

Computational politics, has two branches: applications that build models or knowledge representations from artificial languages, such as rule-based expert systems, and those that begin from natural language.Since most human knowledge is encoded in text, most models will draw from textual sources. If the interpretative process of datamaking is to be explicit and itself an object of study and debate – rather than hidden in forgotten data coding practices – political analysts must use natural-language AI systems to create knowledge representations from the sources texts (Duffy & Mallery, 1986; Mallery, 1988a: 47-66; Duffy, 1988, Duffy, 1989; Alker, Duffy, Hurwitz, & Mallery, 1990). Drawing knowledge directly from long-term social memory (text) into computer systems (Lenat & Guha, 1990), provides one of the few ways to increase the productivity of social science and extend its ability to effectively cope with the expanding complexities of the modern world. Beyond these epistemological considerations, two general propositions about ontology of political phenomena argue that computational politics requires a text modeling capability:

- **Symbolic Basis of Political Rationality:** Individual cognition is grounded in "procedural rationality" (Simon, 1985) symbolically-mediated, functional account of human reasoning processes, where these reasoning processes are conceived as "physical symbol systems" (Newell, 1980, 1989). The purposeful cognitive processes of individuals and organizations, dependent on the linguistic interpretations of meaning and intentions, account for choice and provide the source of political action (Lasswell, 1927, 1935; Lasswell, Leites, *et al.*, 1949; Deutsch, 1953; George, 1959; Pool, Lasswell, Lerner, *et al.*, 1970; Alker, 1975; Habermas, 1979, 1981; Alker, 1984; Dallmayr, 1984; Shapiro, 1984, 1988; Taylor, 1985; Alker, 1986; Mallery, Hurwitz, & Duffy, 1987; Mallery, 1988a; Mefford, 1988b; Duffy, 1987, 1989; Alker, Duffy, Hurwitz, & Mallery, 1990; Hurwitz, 1990).
- **Social Systems as Patterns of Conversations:** The embeddings of people within discourse communities is the main constraint and stimulant of political cognition (Wittgenstein, 1953; Deutsch, 1953; Habermas, 1979, 1981; Dallmayr, 1984; Harre, 1985; Alker, 1986; Mallery, 1988a). Within this framework, political communication and scientific discussions about politics become patterns of arguments. Where consensus is possible, there are shared norms of valid argument. In conflict situations, the validity of argumentative move, and even, facts become problematic.

1.2 Reading the Paper

The paper is organized into two major divisions. Section 2 provides a summary overview of the theory and practice that converts texts into referentially integrated knowledge representations. Next, section 3 reviews the implemented lexical classification system, explaining how to define lexical recognizers, organize them into hierarchies, and apply them to text models. It also reviews the classifications systems developed so far (section 3.3). A discussion of applications draws on the theoretical underpinning of RELATUS and analogizes from traditional content analysis to develop some notions of reliability and validity for semantic content analysis (section 3.4.1). It also proposes other analytical applications, such as precedent logics, modeling differential perceptions, and recognizing argument connectives. The final application domain explains how lexical classification can extend referential coverage in text models, yielding better representations for analysis. The sequencing of this paper is largely suggestive; readers should feel free to read forwards, backwards, and inside-out as necessary. Readers who wish to see examples of text models before reading about the mechanisms that build them, should turn to section 2.5 now.

2 Text Representation in RELATUS

The RELATUS System[3] acquires knowledge by mapping "literal and explicit" texts into a dynamic knowledge representation that captures their referential structure. This representation lends itself to the regeneration of surface texts and to inferential operations associated with common-sense reasoning (Minsky, 1987), such as analogy and generalization. Some major features of RELATUS are:

- **Text Modeling Environment:** It provides a powerful and general foundation for text modeling, including facilities for research and simulation of cognitive processes, whether individual, collective, or abstract. RELATUS was originally conceived as a "social science workbench" to model politically relevant texts and support actual

[3]The RELATUS Natural Language Environment is comprised of a number of component systems, developed since 1983. Gavan Duffy designed and implemented the syntactic parsing system (1.5 megabytes of source code). It includes a syntactic parser with a categorial disambiguator (836 kilobytes), a lexicon acquisition shell (115 kilobytes), and related utilities (585 kilobytes), such as a parse examiner window and a lexicon editor window. John C. Mallery designed and implemented the rest of RELATUS (3 megabytes). This includes the GNOSCERE knowledge representation system (1 megabyte), the reference system (791 kilobytes), the semantic inverter (191 kilobytes), the lexical classification system (231 kilobytes), the sentential constraint poster (292 kilobytes) that quantifies noun phrases and constructs RELATUS constraint descriptions from the output of Duffy's syntactic parser, the syntax interface that allows any parser-generator pair to be used with RELATUS, the question answering system, the RELATUS text analysis mode in the Lisp Machine's ZMACS editor, the belief-system examiner for inspecting semantic representations, and related utilities. Other loadable systems written by Mallery include the prisoner's dilemma phase analysis system (100 kilobytes), a experimental precedential reasoning system, an experimental sentence generator (116 kilobytes) that decodes semantic structures encoded from the Duffy parser, and a lexicon shell for the General Inquirer Lexicon system (205 kilobytes) used to create a RELATUS lexical classifier hierarchy. Mallery also designed and implemented the Feature Vector Editor System (847 kilobytes) that can map feature vector data into RELATUS to build a referentially integrated knowledge representation. The Feature Vector Editor has been interfaced with Duffy's interval time indexation system (149 kilobytes) which incorporates his two dimensional binary tree system (40 kilobytes) and his binary tree system (57 kilobytes). Duffy has been developing a language and theory independent Generic Lexicon Shell with an embedded parser to replace the installed syntax module. The byte size of LISP source for major systems reported here indicates their complexity but does not capture their power or their coding abstraction. RELATUS is implemented in Common LISP and runs on Symbolics LISP Machines. Each author holds individual copyrights (1983, 1984, 1985, 1986, 1988, 1989, 1990) to their respective systems.

or counterfactual simulations (Duffy & Mallery, 1986; Mallery, 1988a; Alker, Duffy, Hurwitz & Mallery, 1989).

- **Domain Independence:** It is a domain-independent AI system because its bottom-up operation relies on the ontology (form) of knowledge representation instead of its specific content (Mallery, 1988a).
- **Phenomenological Representation:** It initially represents meanings by taking snapshots of grammatical relations from sentence parses and referentially linking the constituents across sentences. The initial picture-like, or *eidetic*, representation can receive subsequent, and possibly divergent, interpretations (see section 2.1). Thus, difficult interpretation problems, such as the analysis of belief operators, are deferred.[4]
- **Hermeneutic Orientation:** It is grounded in a constructivist theory of meaning that composes interpretations from an initial eidetic representation rather than a decompositional theory that reduces sentences to putatively universal (but cognitively implausible) semantic primitives (see section 2.1).[5]
- **Conservation of Information:** It does not lose any information present in the original texts; as interpretation proceeds, information is only added. For example, grammatical relations are carried forward from syntactic analysis into the semantic representation.
- **Large Texts:** It is capable of processing large texts (presently, up to several hundred pages) quickly (about 5 minutes to parse and reference 200 sentence texts or roughly 10 pages on Symbolics 3640 Lisp Machines). This means that once text have been converted to "literal and explicit" English (see section 2.2) around 80 to 100 pages per hour can be parsed and represented.

The theoretical ideas that guide the RELATUS project situate the subsequent review of the computational mechanisms to create and manipulate text models for semantic content analysis. The theory of lexical interpretive semantics (section 2.1) grounds hermeneutic interpretation in the phenomenology of syntax. The analysis of reference (section 2.2) suggests how deliberative forms of reference arise from a base in phenomenological reference. These sections adduce theoretical desiderata to distinguish natural-language processing models (section 2.3). A clear understanding of the implemented processing model is crucial for:

- creation and analysis of text models with RELATUS;
- determination of the validity of semantic content analyses;
- recognition of the contemporary limitations of the technology;
- research to extend hermeneutically-oriented text modeling technology.

[4]Knowledge representation schemes based on formal logic (Konolige, 1984; Moore, 1985) require the analysis of belief operators before a specific representation can be defined, and therefore, selected. Thus, these formalisms cannot represent natural language until comprehensive logical analyses are available.

[5]Mallery, Hurwitz, and Duffy (1987) introduce the hermeneutics literature relevant for artificial intelligence.

2.1 Lexical-Interpretive Semantics

Semantic perception is the process of mapping from a syntactic representation into a semantic representation (Mallery & Duffy, 1990). Traditionally, universalist semantics (Katz and Fodor, 1963; Schank, 1972; Schank and Abelson, 1977) advocates determining equivalent meanings (paraphrases) through the decomposition of different surface forms to a canonical semantic form composed of semantic universals, such as "conceptual dependency" primitives (Schank and Abelson, 1977). But, lexicalist semantic theories argue that most meaning equivalences must be determined constructively for specific linguistic communities (or even individual language users) and dynamically for the intentional context. The experimental psycholinguistics supports the lexicalist position (Fodor, *et al.*, 1980; Gentner & Landers, 1985). Referential opacity,[6] which figures centrally in strategic language and decision, poses a debilitating dilemma for semantic universalism because their perceptual apparatus, discrimination nets,[7] provides no means of identifying opaque contexts in order to avoid merging equalities across them without prior deliberation (Maida & Shapiro, 1982; Mallery, 1987). Addition of this capability would require a representation of surface semantics before decompositional perception – but this obviates the need for a universalist representation!

In RELATUS, the construction of semantic representations from canonical grammatical relations and the original lexical items (word stems) is informed by a theory of *lexical-interpretive semantics*. Semantic representations are canonicalized only syntactically and morphologically (partly), not semantically or pragmatically. Lexical-interpretive semantics assumes that meaning equivalences arise because alternative lexical realizations accomplish sufficiently similar speaker goals to allow substitution. A practical argument for dynamically determining meaning congruences is the intractability of a static analysis with sufficient details and nuances to capture subtle variations in speaker goals. The intractability of static meaning equivalences arises from the need to anticipate all possible utterance situations and combinations of language-user effective-histories. [8]

Instead of relying on static equivalences determined in advance, lexical-interpretive semantics requires identification of meaning equivalences at reference time. *Meaning congruence classes* are equivalent semantic representations, conforming to the linguistic experience of specific language users, that could satisfy their utterance-specific intentions.

[6] *Opaque contexts* are linguistic situations where statements, or sentence fragments, are scoped by belief-suspending constructions. These include potentially counterfactual verbs of belief, intention, or request. Verb tense or aspect indicating future occur, subjunctive mood, or conditionals have the same effect. Also, adjectives like 'imaginary' can demand suspension of belief. Opaque contexts require an understander to independently determine the referential status of their contents. In contrast, *transparent contexts* do not require these determinations. In the classic exposition, Frege illustrated referential opacity with his example of the morning star and the evening star, pointing out the need to learn that they are actually one star, Venus. *Propositional attitudes* and *belief contexts* are other names for opaque contexts.

[7] The recent controversies over the inadequacy of discrimination network models of perception (Barsalou & Bower, 1984; Feigenbaum & Simon, 1984) suggest the need for a new account of perception. The constraint-posting model of semantic perception is such a new perceptual mechanism with better characteristics (Mallery & Duffy, 1990).

[8] A language user's *effective history* is his personal experience including the cultural and linguistic traditions inherited according to his position in society.

Lexical-interpretive semantics can avoid the distortions due to exclusive reliance on a static analysis of meaning equivalence because it selects equivalences from synonym or paraphrase congruence classes determined on the basis of dynamically changing, intentional contexts of language use. Although the theory calls for dynamically determination of meaning equivalences at reference time for historical, individual speakers, the present practice in RELATUS relies on a universal syntax for idealized language users, and to the extent implemented, static meaning congruences for belief systems with specific background knowledge (see section 2.3).

Since RELATUS retains the original lexical items from sentences, the resulting semantic representations are lexicalist and referential opacity is the norm (Maida & Shapiro, 1982). Semantic perception does not substitute equals for many cases.[9] Substitution of equals across opaque contexts never takes place; instead identity relations may be asserted after valid equalities are determined. Although the determination of dynamic meaning congruences may require attention to belief contexts, at least, the belief contexts and their contents are already represented so that a reasoner may consider them!

Lexical-interpretive semantics is hermeneutic because the theory emphasizes interpretation based on the effective-histories of language users and the intentional structures of communicative situations.[10] It is aligned with the phenomenological hermeneutics (Ricoeur, 1975) because both emphasize the evolution of meaning through metaphorical innovation in language and identify metaphor as a primary source of polysemy. Both begin interpretation from an eidetic level of representation, but lexical interpretive semantics has received computational rendition in the RELATUS system that reaches the level of eidetic representation, and even, somewhat beyond. After the syntactic analysis, grammatical relations derived from deep structure and noun-phrase quantifiers computed from surface structure are posted as declarative constraints to construct a sentential logical form. The reference process resolves the constraint description for a specific the semantic representation, completing the eidetic perceptual process. This perceptual process does not involve inferences or non-syntactic equivalence substitutions; explicit descriptions are simply matched against memory. Thus, lexical, constraint-based graph matching forms the ground for more complex deliberative processes of a more general and open-ended hermeneutic interpretation.

The lexical-interpretive grounding of RELATUS requires a lexical classifier to constructively classify configurations of lexical semantic relations. Here, classification finds categorically equivalent nouns and verbs, whereas decompositional semantics would have reduced different words to the same internal primitive as it merges equivalences during input. For example, the lexical classifier can recognize and constructively categorize physical movement verbs (conceptual dependency's "PTRANS" primitive). Once classified, the original statement remains for future reinterpretation and no information is lost. Lexical classification is the most basic inference mechanism in a lexicalist approach to semantic representation. Indeed, lexical classification in this role of constructive classification becomes the first step in hermeneutic interpretation. It is hermeneutic is because instances found for categories depend on the textual history of the understander,

[9] Opaque adjectives and adverbs are not detected because the present lexicon (Duffy, 1987) does not contain this information.

[10] In practice, individual histories are not available, so analysts must reconstruct in a psycho-biographical mode relevant histories as information resources allow. Earlier forms of cognitive modeling in political science, such as cognitive mapping, faced a similar difficulty.

which may vary from hearer to hearer. In theory, lexical categories should be induced for each understander's history(Piaget, 1979). Although different understanders should have some overlapping categories, there should also be differences following from experiential variations.

2.2 Immediate vs. Deliberative Reference

Even though philosophy of mind has long distinguished internal (private) and external (public) reference (Russell, 1948: 114, 224-231), contemporary linguistics has often emphasized external reference, or the correspondence of terms (words, sentences, collections of tokens) to the "real" (external) world (*e.g.*, Lyons, 1977: 174-230; Bach & Harnish, 1979; Barwise & Perry, 1983). Similarly, hermeneutic social science, such as the communitarian/peace research paradigm in international politics, focuses on the internal reference of political actors, whereas rationalist social science, such as realist theories of international politics, remain committed to external reference. Any computational model of politics, such as political applications of the RELATUS system, must rely on internal reference to create a coherent model in the computer. For these internalist models, external reference is the difficult problem of model validation, or identifying the correspondences between the computer model and the external world.

When people read texts and construct their internal models, they rely on many distinct modes of inference to locate referents. Since these inferences may also draw on arbitrary knowledge accumulated over a lifetime of social relations and social communications, effective study of the internal reference conundrum requires a division into simpler, "nearly decomposable" (Simon, 1962) components.

The theory behind the RELATUS reference system envisions two categories of internal reference:

- **Immediate Reference** simply looks up literal relations from *explicit* representations in memory.
- **Deliberative Reference** relies on reasoning, which may be arbitrarily complex, to select possible referents. Deliberation becomes necessary because the intended referents are implicit, demanding inferences that draw on background knowledge to connect the explicit text coherently. However, deliberation presupposes the ability to identify and manipulate the terms involved in reasoning. Thus, *deliberative reference necessarily builds from immediate reference.* The need to deliberate about referents is certainly signalled by contradictory, incommensurate, or non-existent referents as well as violation of selection constraints for word senses, semantic or pragmatic ambiguity, or conversational inadequacy (Searle, 1979; Levinson, 1983: 157-158). But it also seems required to find coherent interpretations of discourse situations (Hobbs, 1979, 1986).[11]

[11] Roger Hurwitz (personal communication, Spring, 1989) notes a deconstructive component to deliberative reference. Readers may understand the intended referents of an author, and may additionally, recognize unintended associations. These additional interpretations do not restrict reference but they do extend and situate it for the reader.

The dependence of reference on inference is the criterion that distinguishes immediate from deliberative reference. Inferential dependence raises the possibility of its invalidation by future information. The architecture can identify this contingency only if relations supporting the inference are recorded. Then, the system can recover by either showing that different support relations justify the inference or retracting it along with any associated references or dependent inferences.

Natural language understanding demands many modes of inference and learning in deliberative reference. Thus, after detecting the failure of immediate reference, deliberation must select an inference strategy to resolve the reference. Lexical classification of the semantic structure retrieved or created by immediate reference is a first deliberative step that constrains the selection of subsequent inference strategies. For example, classification might indicate the violation of a verb's selection constraints and call forth an inference strategy to resolve this metaphorical usage.[12]

Deliberative reference can be decomposed along several axes.

- **Teleological/Non-teleological:** *Teleological reference* requires imputing plans to purposeful agents[13] and reasoning about their plans in order to locate referents while *non-teleological reference* does not. Teleological reference includes *conversational implicature*, implicit premises inferred on the basis general conversational principles (Grice, 1975, 1978; Levinson, 1983: 97-166).

- **Deductive/Hypothetical:** *Deductive reference* can locate referents by necessary reasoning whereas *hypothetical reference* requires hypothesis generation. Hypothetical reference decomposes into *stereotypical reference*, *inductive reference*, and *abductive reference*. Inductive reference relies on inductive generalizations. Abductive reference utilizes analogy, synecdoche, or metonymy to propose hypotheses.[14]

- **Literal/Figurative**: Metaphor and irony are the major figurative forms of abduction. The literal side might be best thought of as 'dead' metaphors of forgotten origin. The literal/figurative distinction highlights the poesis arising as *figurative reference* interprets novel configurations of familiar tokens. While new hypotheses may be adduced by necessary/possible reference, new senses emerge in figurative reference.[15]

---

[12]Deliberative reference remains a largely theoretical goal in the present system.

[13]Chapman (1985) shows that planning is both undecidable and computationally intractable in the general case. To the extent that plans are imputed to agents on the basis of stock knowledge or stereotypes explicitly represented in memory, these difficulties can be avoided. Such a strategy may sidestep the computational difficulties only at the expense of completeness. Mallery (1987, 1988a) discusses this problem as it appears in strategic language and thought and as it applies to AI models of international politics.

[14]Readers unfamiliar with the concept of abduction should consult Peirce (1901), Rescher (1978), Charniak and McDermott (1985). Peirce distinguishes two types of abduction. *Perceptual abduction* is hypothesizing and recognizing objects from perceptual sources. *General abduction* is hypothesis formation in non-perceptual, internal thought.

[15]For an excellent overview of the metaphor literature, especially the debate over the interaction (the directionality of mapping), see Ricoeur (1977). Lakoff and Johnson (1980) propose metaphor as a fundamental cognitive process. Doug Lenat (personal communication, December, 1989) reports that

AI systems require all these varieties of reference, including supporting inferential machinery, to approach human capabilities. Thus, these distinctions suggest a *referential competence* found in natural language understanding.[16] Since all categories of deliberative reference presuppose immediate reference, a performance model of reference must first account for immediate reference. Afterward, it must allow incremental extension to gradually bootstrap more complex kinds of deliberative reference from simpler ones.

2.3 Models of Natural Language Processing

Along a referential bootstrap sequence, natural-language processing models range from *single-sense, literal and explicit processing* through multi-sense, literal and explicit processing to partially deliberative, and ultimately, fully-deliberative or AI-complete processing.[17] *Single sense* refers to word usage where only one meaning of the word is used within any of its parts of speech (*e.g.,* noun, verb, adjective). *Multi-sense* processing requires the ability to discriminate different senses of words within part of speech and recognize equivalent paraphrases. *Literal language* refers to text in which no metaphors or other tropes appear and all words are used according to a single definitional authority, such as an analyst or a text producer. *Explicit language* contains no implicit premises or referents that require inferences drawing on background knowledge. Because the referential connections between sentences are explicit, the coherence of the text becomes manifest.

Immediate reference accounts for the most basic model, single-sense, literal and explicit processing. Constraint-directed graph matching finds the correspondences from sentences to semantic memory. Word senses cause no confusion because only one sense appears for each part of speech. The absence of tropes and implicit referents defers some of the difficult problems of reference.

The move to multiple word senses begins by using lexical classification to recognize and label different senses. It continues by using meaning congruences to recognize dynamically equivalent paraphrases for references across sentences. The present version of RELATUS is now approaching these capabilities. The reference system has the ability to recognize nominalizations of verbs. The lexical classifier uses selection constraints to identify different senses of words. Identifying equivalent paraphrases and differentiating word senses shades into deliberative reference as the application of selection constraints and cross-graph mappings demand inferences. Although this model is far from unrestricted natural language, it is nevertheless a useful beginning that allows interesting representations for text to be created and used in reasoning and learning applications.

The important scientific point is that the distinction between immediate and deliberative reference provides a theoretical framework to decompose AI-complete natural language understanding into a bootstrap succession of simpler, independent extensions

fixed interpretive templates for standard metaphors cover a surprising number of cases arising in his project to develop encyclopedic knowledge bases (Lenat & Guha, 1990).

[16]This referential competence sketches a foundation for a computational theory of something like a "universal pragmatics" or a communication theory social action (Habermas, 1979, 1981).

[17]By analogy to NP-completeness in complexity theory, Fanya S. Montalvo coined the term "AI-complete" to denote a computational problem whose difficulty is equivalent to solving the central AI problem, *i.e.* making computers as intelligent as people.

that lead cumulatively to better processing models. Pedagogically, careful specification of the decomposition for a particular implementation allows users to anticipate the behavior of a natural language system, and not be surprised by "break downs" (Winograd & Flores, 1986) when tasks exceed the model specification. There is no claim that a system "understands" natural language; rather grammatical, referenetial, and inferential competence gradually evolves from highly-restricted toward ever less-restricted language as incremental research discovers how to extend the implemented model.

2.4 Creating Text Models

The RELATUS implementation has been guided by the design goal of felicitous representation of large texts for unrestricted content domains. Domain-specific or other *ad hoc* strategies inconsistent with this goal were eschewed. RELATUS gains broad coverage and domain-independence from a bottom-up strategy that combines a general syntactic analysis with a constraint-posting reference system to create large, referentially integrated semantic representations.[18] The major components that make this possible will now be reviewed.

2.4.1 Knowledge Representation

Unlike conventional knowledge representation schemes, the GNOSCERE subsystem provides *belief system* objects, which are each instantiations of a knowledge-base object-class, handling a set of generic operations.[19] All processing in RELATUS is organized around belief systems. Their major operations include syntactic analysis using the current parser (see section 2.4.5), semantic reference using the constraint-interpreting reference system, sentential constraint-posting, and question answering. Other operations include tracking

[18]Winston (1980; 1984) and Katz (1980; Katz & Winston, 1982) used Katz's deep structure transformational parser to build representations in Winston's ternary-relation frame system. RELATUS is a descendant of that system. RELATUS differs, *inter alia*, from the Winston representation system because it adopts a bidirectional interpretation for ternary relations. This makes possible a reference system, and thereby, facilitates production of referentially integrated knowledge structures. It also makes possible a semantic inverter to walk graphs and perform activities, such as constructing constraint descriptions for question answering or lexical classifiers, displaying graph structures in the belief system examiner, and generating sentences. Additionally, RELATUS includes user-interfaces that make processing large texts practical. Unfortunately, Duffy's (1987) doctoral dissertation neither details the similarities between the Katz and Duffy parsers nor contrasts the Duffy parser to other efforts in computational linguistics. Originally, the Winston-Katz system reasoned analogically from small textual inputs. More recently, Katz (1988) proposes his successor system for indexing and retrieving English knowledge. This proposal would be more convincing if he presented a general reference system, a semantic perception model, and discussed completeness, correctness, efficiency of retrieval. Syntactic coverage is difficult to assess for both the Duffy and Katz parsers because their procedural implementation restricts the access of non-implementors to their grammars. As open, easily-revisable parsers based on declarative grammars become the norm, procedural parsers will obsolesce.

[19]RELATUS has uses the Symbolics Flavor System extensively for object-oriented programming. The implementation will be revised as the new standard for LISP, the Common Lisp Object System (De Michael, 1989), becomes available.

the dependencies of semantic structures to their source sentences, displaying knowledge structures, processing user-defined directives (actions for the system to take when given imperative sentences by the user), and invoking pre-defined "if-added" procedures whenever trigger relations are created in a belief system. Their semantic representations are a special class of graph structure (or semantic network) constructed from ternary relations, or labeled binary relations. An important feature of ternary relations is that they are arbitrarily expressive.[20] Different interpretive regimes can be concurrently modeled by instantiating multiple belief systems and directing them to parse and represent different texts. Within belief systems, memory is implemented as a "society" of communicating graph-node agents (Minsky, 1987), which support over five hundred operations. Belief systems maintain both short-term and long-term societies of graph-node objects, and manage the locality of these knowledge structures to improve performance for large scale applications.[21] The process of text modeling is mapping surface text into belief-system knowledge representations.

2.4.2 Constraint-Interpreting Reference

A constraint-interpreting reference system[22] (Mallery, 1990) is an interpreter and an extensible set of constraints. The reference system *finds* and *creates* graph structure in semantic representations, serving a function analogous to LISP's intern.[23] The constraints constitute a declarative language for describing graph structures. Collections of constraints are bundled in units called *reference specifications* (or REF-SPECs). These structures are actually trees of message-passing objects that can be associated with a belief system. Each ref-spec hierarchy uses the self-indexation capabilities of belief systems to generate possibilities and the connectivity of their knowledge graphs to select the correct graph structures for retrieval.[24] This self-indexation is based primarily on the token-type[25] of lexical markers and ternary relations. Alternate graph traversals leading to desired graph nodes, such as classification (the class membership of nodes) or similar

[20]Since natural languages allow us to talk about just about anything, including this sentence, the safest approach is to use an arbitrarily expressive representation and restrict its expressibility only as necessary for specific applications.

[21]While working with the Winston-Katz system in 1980-83, the author could not represent texts larger than about five pages on early Lisp Machines because of "page thrash." Thrashing occurs when a computer with a virtual memory architecture (real memory and swapping to disks) cannot swap in the task into real memory because the task elements are two widely scattered in memory or too large for real memory. One of the initial motivations for beginning work on RELATUS during the summer of 1983 was to develop a knowledge representation system that could avoid thrashing yet represent enough text for non-trivial social-scientific applications.

[22]The present reference system is a complete reimplementation that overcomes serious limitations of the first version developed in 1983. The reimplementation began in the summer of 1985 and the new version was installed during the summer of 1986.

[23]The intern function returns the symbol object given its print name and package.

[24]Unlike many frame representation systems, GNOSCERE belief systems support *complete* indexation of their graph representations from *any* topological orientation. The space complexity of this indexation method asymptotes to twice database size.

[25]The graph node representing the token class is the *token-type* of all occurrences of the token. The tokens coming from natural-language input are morphologically normalized when the parser removes differentiating suffixes.

abstractions, *ipso facto* extend the indexation. Instead of relying on a fixed indexation system (based on a necessarily incomplete set of fixed keys), this representation system is *fully indexical*. Indeed, the indexicality evolves as new tokens are added or new classifications are performed. This makes the reference system complete, ensuring that it will find all graph nodes denoted by a set of constraints.

This reference system finds graph nodes satisfying any ordinary constraint specification, which include no inferences, in time independent of database size. To match a ref-spec, the system:

- **Orders constraints** according to their "pruning power" – an often precise factor that reflecting the number of nodes satisfying the constraint;
- **Generates a possibility space** from the constraint satisfied by the fewest nodes;
- **Applies Constraints**, previously ordered, to efficiently prune the possibility space, yielding nodes satisfying the specification.[26]

The general principle that makes constraint-interpreting reference very efficient is the use of a declarative representation of the graph matching task to identify and side-step, or postpone, difficulties instead of blindly stumbling into them.[27]

This system is termed a reference system because it aims to computationally model internal reference. The extensible constraint language allows constraints to be defined for deliberative reference operations as the inferential capabilities to support them become available. It also allows users to define specialized constraints for their applications. The constraint language presently contains about 150 types of constraints, a number of which are lexical predicates for use in sentential reference. A meaning congruence facility currently finds referents by constructing alternate reference specifications for certain regular cases of agentive nouns (e.g., murderer) and simple derived nominals (e.g., destruction). In general, all other systems that access the belief-system knowledge bases must express their database operations as ref-specs.

2.4.3 Syntactic Analysis

The Duffy parser[28] (Duffy, 1987) is a deep-structure transformational parser (Chomsky, 1965; Jackendoff, 1972) for English syntax.[29] The parser produces a directed cyclic

[26]The reference system does not backtrack during the constraint application process. The analog of backtracking is the number of unsuccessful possibilities that appear in the initial possibility space.

[27]Difficult constraints such as ones that attempt to prove theorems about a node or one that applies some exponential algorithm are possible because the user has the freedom to define new constraint types. However, these constraints would have a poor constraint factors, and therefore, would be applied only after more efficient constraints had pruned the possibilities. In general, the time complexity of a reference specification is additive in the time complexity of each constraint.

[28]This discussion draws from earlier descriptions by Duffy (Duffy & Mallery, 1986: 21, 22; Duffy, 1987).

[29]Because sentential look-ahead is unbounded, the parser is not deterministic in the sense of Marcus (1980). However, because operations terminate at clausal boundaries, the parser is *effectively* determin-

graph which describes the syntactic constituents of a sentence. Interwoven within this cyclic graph are structure-sharing trees (directed acyclic graphs) which describe the syntactic deep structure and syntactic surface structure of a sentence. Components are included within the parser for lexical insertions of ellipses, punctuation handling, possessive handling, English-Arabic numeral translation, and intrasentential anaphora resolution. Crucially, the parser outputs both surface and deep structure representations. Surface structure is required to determine correct quantifier scoping while deep structure or canonical grammatical relations are required for technical reasons related to sentential constraint posting, coherent semantic representation and efficient reference (Duffy & Mallery, 1984; Mallery & Duffy, 1990; Mallery, 1990). The object-oriented implementation of the parser has not only allowed greater complexity while maintaining simplicity but it has also enhanced modularity and facilitated the interface to the sentential constraint poster (section 2.4.4). The procedural implementation of the parser makes it very fast, and therefore, suitable for parsing large texts. A key innovation that makes transformations tractable involves maintaining clausal constituents in a list so that movement merely involves copying and reordering the list rather than exponential, tree-descent copying of parse structure.[30] The parser has quite broad coverage of English grammar.[31]

The parser incorporates a categorial disambiguator (Duffy, 1986) that identifies the correct part of speech for each word in a sentence by means of a constraint propagation scheme (Waltz, 1975). When an ambiguous category is encountered, the disambiguator exploits constraints propagated from its neighbors. Thus, the successful disambiguation of one ambiguity in a sentence propagates additional constraint which may be helpful in disambiguating other ambiguities. Unlike some parsers that handle multiple parts of speech, the disambiguator and the syntactic analyzer are functionally independent. This allows the disambiguator to run before the parser attempts to analyze the grammatical structure of each sentence. Thus, the disambiguator makes parsing more efficient because it allows the parser to ignore many alternative parses that would otherwise arise from part of speech ambiguity. Of course, part of speech disambiguation allows the same word to appear in differing roles (e.g., as a verb and a noun). Thus, the ability to use the word senses associated with different parts of speech is purchased without recourse to a semantic sense disambiguator. Categorial disambiguation is therefore an effective means of increasing parser efficiency and linguistic coverage for a small computational expense.

2.4.4 Sentential Constraint-Posting

A *sentential constraint poster* converts the output of the parser (parse graphs) into reference specifications, which mediate the mapping from syntax to semantics. The conversion utilizes both surface and deep structure from the syntactic analysis. While noun-phrase quantification (Mallery, 1985) is analyzed at surface structure, deep-structure grammat-

---

istic. It remains a polynomial LR(k,t) algorithm, although k is variable, not constant. See (Berwick & Weinberg, 1984: 192) for a discussion of the time complexity of LR(k,t) parsers.

[30] Katz (1980; Katz & Winston, 1982) applied this computational principle in his parser, demonstrating that deep-structure transformational parsing was tractable, and thereby, refuting the intractability argument against deep structure (Winograd, 1971: 197).

[31] The precise coverage is not documented and varies according to the mix of bugs that may lurk in the code at any time. But, impressionistic estimates by the author and others suggest that the Duffy parser has broader coverage than the Katz parser, and probably, most extant research parsers.

ical relations are used to construct sentential reference specifications. This process is called sentential constraint-posting because the various pieces of information comprising the sentential description are incrementally collected in a single sentential reference specification. These specifications are constructed during a depth-first walk of deep structure, in which certain constituents displace their constraint descriptions to others. Relative clauses and prepositional phrases displace to the noun or verb the modify just nominal or adjectival modifiers displace to nouns and adverbial or clausal modifiers displace to verbs. Bottom-up, *a priori* grouping techniques for syntax ensure the correct placement of independent references for semantically individuated entities. Thus, as displacement compresses the original syntax into a smaller reference specification, it also reduces referential ambiguity by increasing constraints on independently-referenced constituents. When complete, the reference specification is referenced in a belief system. In contrast to earlier approaches, which performed incremental references before building complete sentential descriptions, sentential constraint-posting exploits a much more effective "wait and see" strategy.[32] Using both deep structure and constraint posting eliminates backtracking in both constraint posting and sentential reference (Duffy & Mallery, 1984; Mallery & Duffy, 1990; Mallery, 1990). As successive sentences of a text are referenced, structures referring to existing representations are found and new structures are added for those with no existing counterpart. In this way, the system builds up a referentially integrated semantic representation, a belief system graph structure that correctly captures references to the same entities across sentences. This process of *intersentential reference* makes possible the construction of referentially coherent text models on a large scale.

2.4.5 Syntax Interface

The only language or (linguistic) theory dependent components of RELATUS are the syntactic components, including the preparser, the parser, the lexicon, the sentential constraint poster, and the syntactic part of the sentence generator. Parsing sentences syntactically is really decoding syntax into ref-specs while generating sentences is encoding ref-specs into syntax.[33] Thus, it makes sense to consider all the submodules of syntax as one big module and to group them together so syntax modules can be easily switched. This allows use of the general RELATUS environment to parse, represent, and generate different languages, or merely use of other English parser-generator pairs. The recent installation of a new Common Lisp version of the Duffy parser provided the opportunity to encapsulate the entire interface to syntax in a single interface object with a declarative protocol. Since certain tokens and predicates used in the semantic representation must necessarily reflect the language, and perhaps the linguistic theory, the syntax interface contains slots for all special tokens. Because RELATUS now has a declarative interface to all syntax operations, including parsing, generation, and access to the lexicon, any parser-generator pair that can answer to the protocol can provide the syntax service for RELATUS. Naturally, the parser-generator pair needs to support a parser capable

[32] To improve apparent psychological realism, substructure of sentences may need to be referenced in order to resolve semantic ambiguities in clausal and prepositional attachments.

[33] Actually, the syntactic configurations, grammatical relations, carry forward into the semantic representation through the ref-specs. But, decoding could go directly back to deep structure because the generation direction does not pose a graph matching problem.

of tractably producing sentential reference specifications,[34] and preferably, a generator, driven by the same grammar, to invert structures produced by the parser. The syntax interface spans all of the syntax related commands in the entire RELATUS environment so that a user can use all the same generic commands for all different parser-generator pairs. For semantic content analysis, this interface means that the facilities of lexical classification can be applied to any language for which there is a RELATUS compatible parser.

2.5 The Text Processing Cycle

This section illustrates the creation referentially-integrated text models over three levels of linguistic processing:

- **Syntax:** Sentence strings are syntactically analyzed to produce surface structure and necessary deep structure.
- **Logical Form:** The lexical content and structural relations of syntax are expressed as sentential reference specification in the constraint language.
- **Semantics:** The reference system creates referentially integrated graph representation as resolves the constraint description, looking for matching antecedents but creating new structures when none are found.

The syntactic analysis process begins when a belief system parses a sentence stream from an editor buffer or text file. Before proceeding, the belief system determines the current *deictic context*[35] either from a global values or from a more specific set of parameters associated with the sentence stream. It invokes a preparser that accepts sentence from the stream, produces a preparsed sentence list, and feeds it along with the deictic context to the parser.[36] Preparsing expands contractions, evaluates lisp forms, and removes comments in the text stream. The parser returns an analyzed sentence object and the belief system references the sentence. This first invokes the sentential constraint poster to convert the parse to a ref-spec. Next, the ref-spec references itself with respect to the belief system. The belief system repeats the procedure sequentially and incrementally until it reaches in the end of sentence stream – a stream to any text source, including files, buffers-regions, or the user.

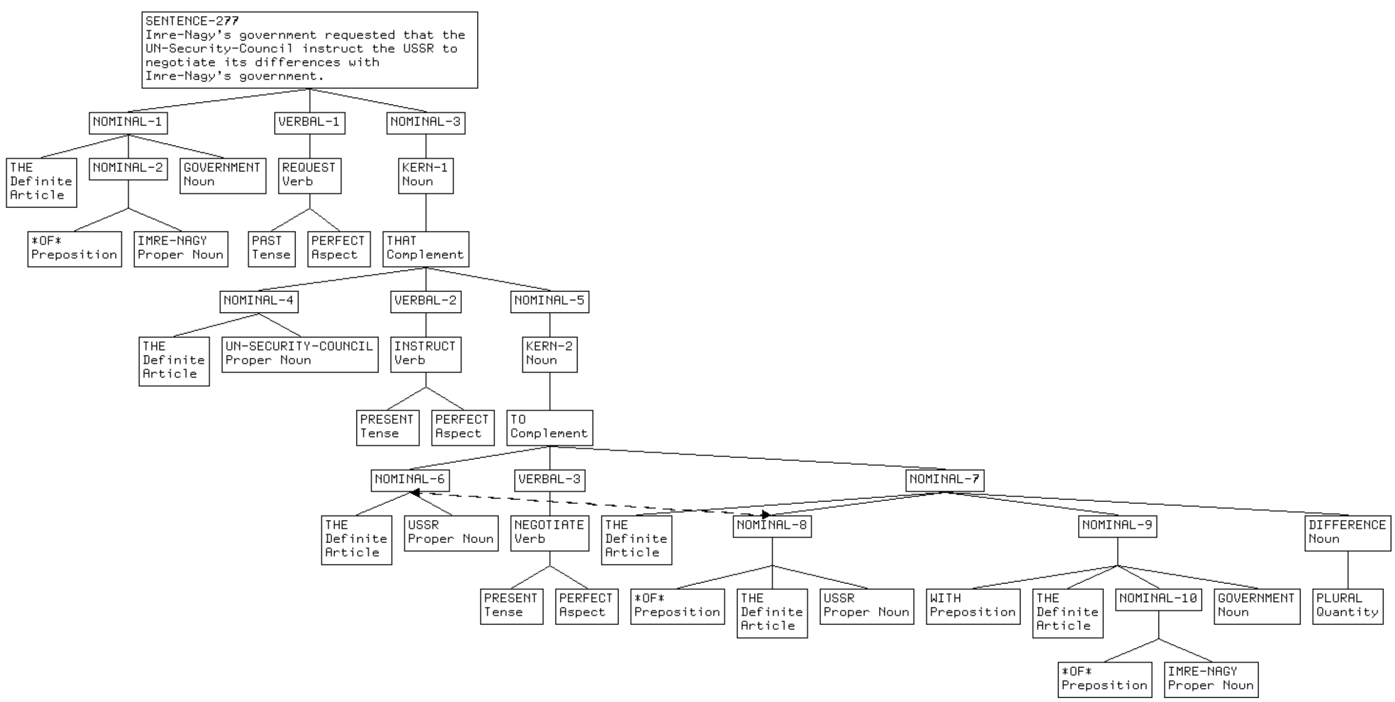
Figure 1 shows a syntactic parse of a sentence from a RELATUS parsable text about the 1956 Soviet Intervention in Hungary (section 2.6.1). The dashed arrow connecting 'nominal-6' to 'nominal-8' indicates the intrasentential coreference found when

[34] The analysis in (Duffy & Mallery, 1984) suggests that this might not be easy for all linguistic theories.

[35] The deictic context of text is a collection of indexicals associated with a text that includes, for example, the source, the recipient, the coding location, the reference location, and the coding time. These indexical situate the text and allow indexical pronouns such as "me," "you," "here," "there," and "now," to refer. This facility constitutes the outside of a context mechanism. It was designed according to categories suggested by Levinson (1983: 54-94).

[36] The installed preparser is a reimplementation by the author of an earlier design by Duffy. Duffy has yet another implementation but it does not support incremental sentence parsing from a stream.

Figure 1: The syntactic parse of a sentence from a RELATUS parsable text about the 1956 Soviet Intervention in Hungary.



the parser resolves the possessive pronoun, 'its,' to 'USSR.' This equality is then carried over through constraint posting.

Figure 2: The semantic structure created from the syntactic parse of figure 1.

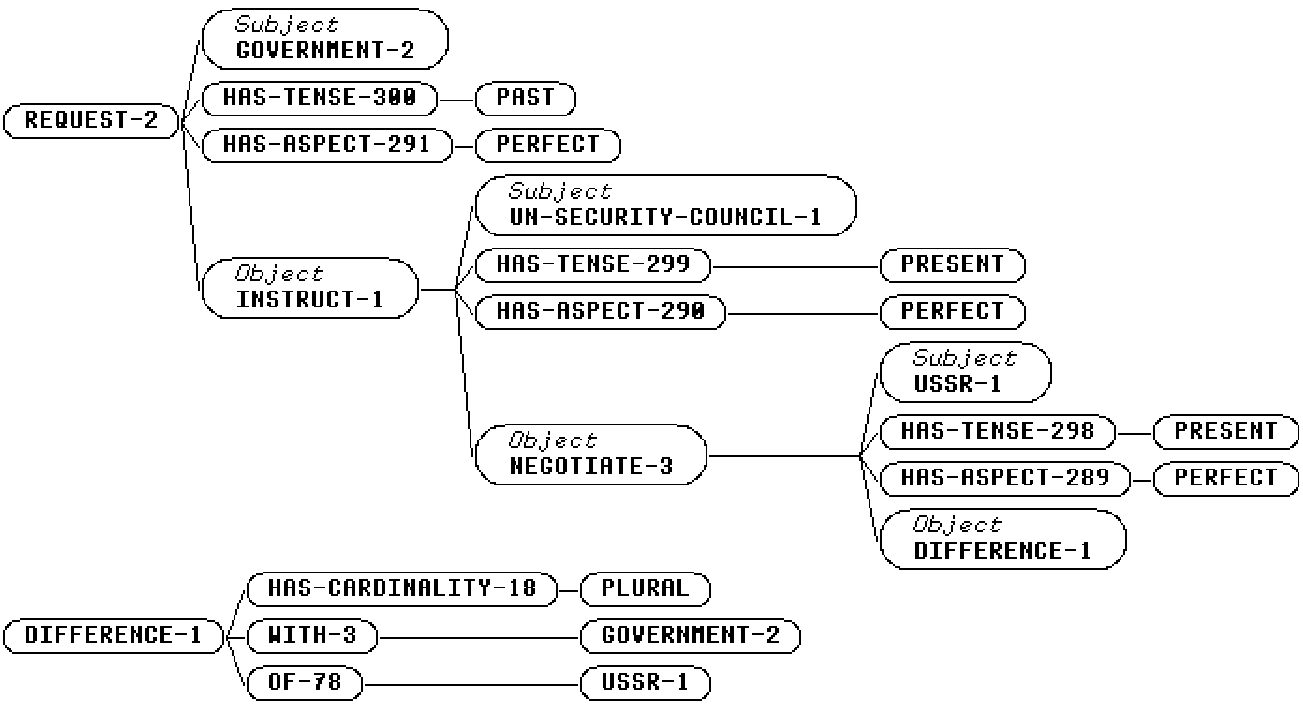
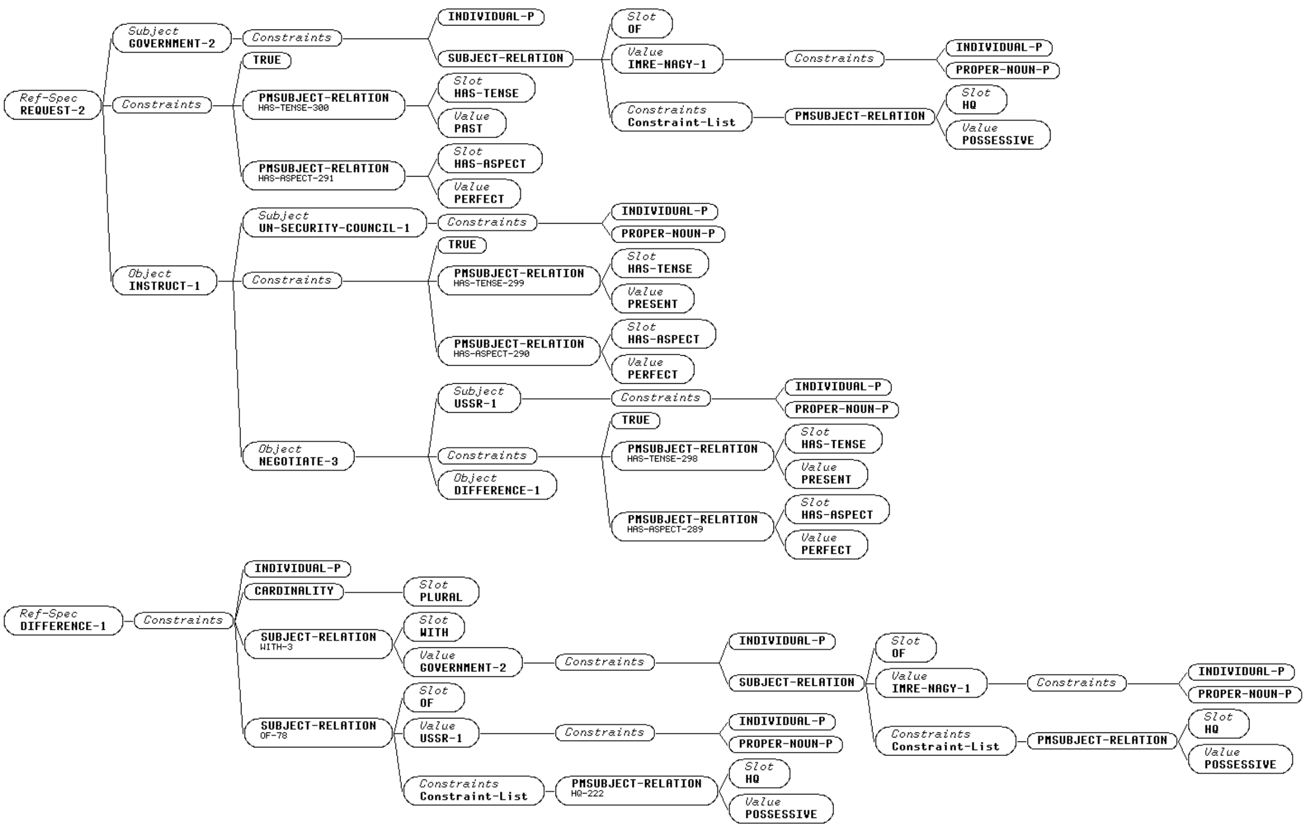


Figure 2 shows the isolated semantic structure to which the reference system resolves the sentence. The toplevel relation reads: 'government-3' 'request-2' 'instruct-1.' Similarly, 'instruct-1' reads 'UN-Security-Council-1' 'instruct-1' 'negotiate-3.' Notice that the object of 'negotiate-3' is 'difference-1' and it has a *subject relation* which is read 'difference-1' 'of-79' 'USSR-1.' But, the subject of 'negotiate-3,' 'USSR-1' has an *object relation* which is read backwards 'difference-1' 'of-79' 'USSR-1' for human intelligibility. In each case, 'of-79' is the same relation, but it is accessed from different directions. When it is an object relation, it appears in italic font to distinguish it from its display as a subject relation in normal font. This illustrates the bidirectional interpretation of GNOSCERE representations.

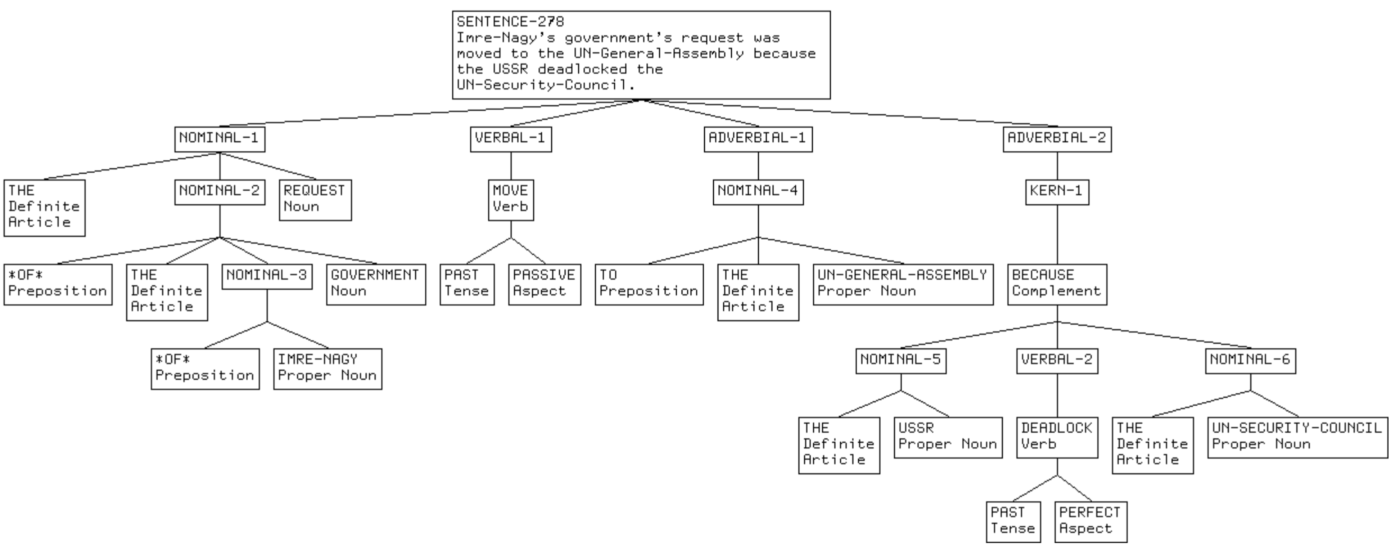
Figure 3 shows the ref-spec that mapped between the syntactic parse in figure 1 and the semantic structure in figure 2. The gennames (word + number) scattered around the ref-spec hierarchy correspond to the gennames in the semantic representation of figure 3. These correspondences indicate the resolution of the constraints in the ref-spec. The top (left) ellipse is the ref-spec for the main or matrix relation of the sentence.

Figure 3: The sentential reference specification that maps the syntactic parse of figure 1 into the semantic structure of figure 2.



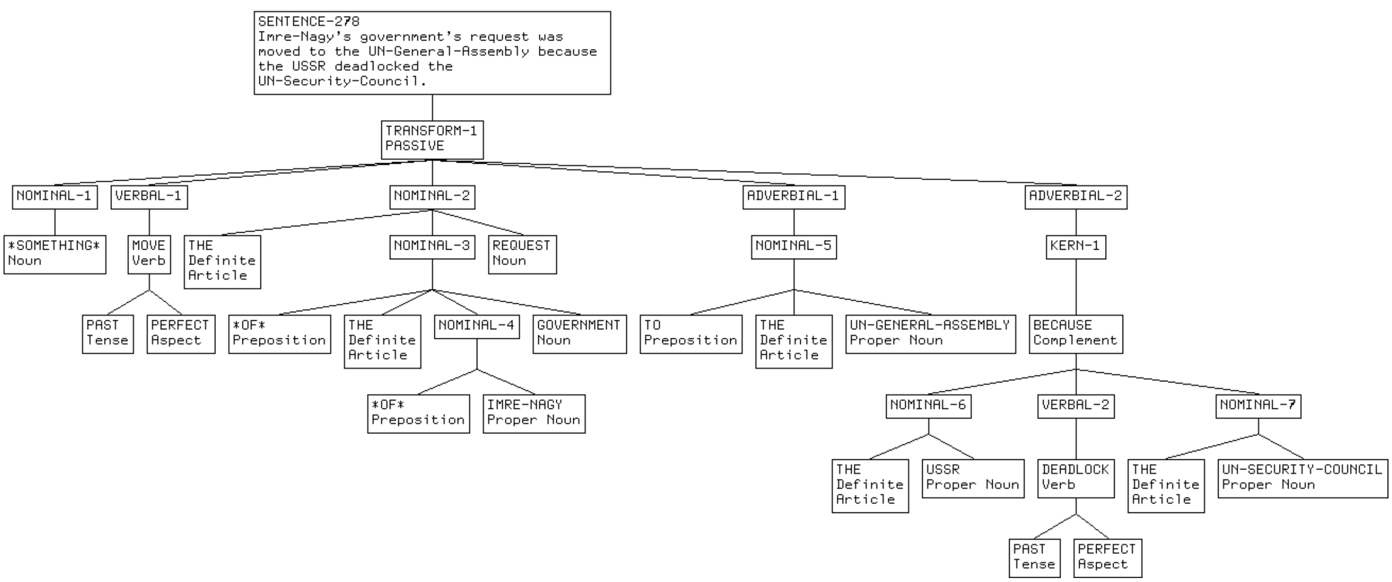
The gennamed word inside the ellipse, 'request-2,' denotes the graph node to which the ref-spec resolved in the belief system. The ref-spec has a *subject* ref-spec, resolved to 'Government-2' and an *object* ref-spec, resolved to 'instruct-1.' Between those ellipses is one labeled, *constraints*. Below are some constraints on request, such as the mandatory constraint *TRUE*. Below *TRUE* is a *PMSUBJECT-RELATION* with the arguments 'has-tense,' 'past.' This constraint says to prefer (P) a node for a 'request' that has a subject relation, 'has-tense,' whose object is 'past.' If the successful referent does not satisfy this preference, the constraint mandatorily (M) assures it by creating a subject relation for the referent. 'Government-2,' the object ref-spec for 'request-2,' has a constraint *INDIVIDUAL-P* which requires the referent to be an individual rather than a universally quantified node. An example of an ordinary mandatory *SUBJECT-RELATION* can be found as a constraint of 'negotiate-3.' It denotes a prepositional link to a ref-spec for 'government-2' which in turn has a *subject-relation* constraint, resolving to 'of-79,' to 'Imre-Nagy-1.'

Figure 4: The surface structure of a syntactic parse that refers intersententially to the semantic structure for 'request-2' in figure 2.



The sentence examined above appears as the first instance of 'request-2' in figure 11. The figure shows intersentential references to 'request-2' in three later sentences, just as figure 10 shows that these mentions refer correctly to the same 'request-2' even though an earlier 'request-1' could be confusing. Figure 4 shows the surface structure parse for the last of these sentences. Because the sentence is passive, the parser produces the deep structure analysis in figure 5 using a passive transformation. The nominal for the request now appears in the object position and '*something*' appears in the subject position – because the verb, 'move,' takes an implied subject. However, the

Figure 5: The deep structure created by a passive transformation of the syntactic surface structure in figure 4.



adverbial 'because' complement remains unchanged. Figure 7 shows the syntactically canonical semantic structure created in the reference of the sentence. The previously mentioned 'request-2' appears as the object of the new relation, 'move-1,' which has a new '*something*-3' as its subject. 'Move-1' also has a subject relation, 'to-11' to 'UN-General-Assembly-1' illustrating the retention of surface prepositional relations. The clausal complement, 'because,' appears as the italized 'cause-72,' is an object relation for 'move-1' representing an antecedent subject, 'deadlock-1.' As the subject and object were transposed by the encoding in the sentential constraint description of a object relation rather than a subject relation for the 'because' complement. Figure 6 shows the constraint, *PMOBJECT-RELATION* for 'cause-72,' on the ref-spec for 'move-1.' Another interesting element in figure 6 is the *meaning congruences* just above the constraints for the ref-spec 'request-2,' the object of 'move-1.' Since the sentence contained a reference to 'request' as a noun and 'request' was known to be the nominalization of a verb, the reference system automatically constructed some constraints to look for any relational mentions of 'request' that might satisfy the verbal projections of the constraints for the noun. The meaning congruences were responsible for finding the 'request-2' referent of the noun phrase in this case. In contrast, the constraint structure for 'deadlock-1' is relational and it also retrieves a relational referent from an earlier sentence.

Figure 6: The sentential reference specification that combines grammatical relations from deep structure (figure 4) and noun-phrase quantification from surface structure (figure 5) to reference the semantic structure in figure 7. The reference system automatically creates the meaning congruences that look for the verb form of 'request,' the nominalized deep sentential object.

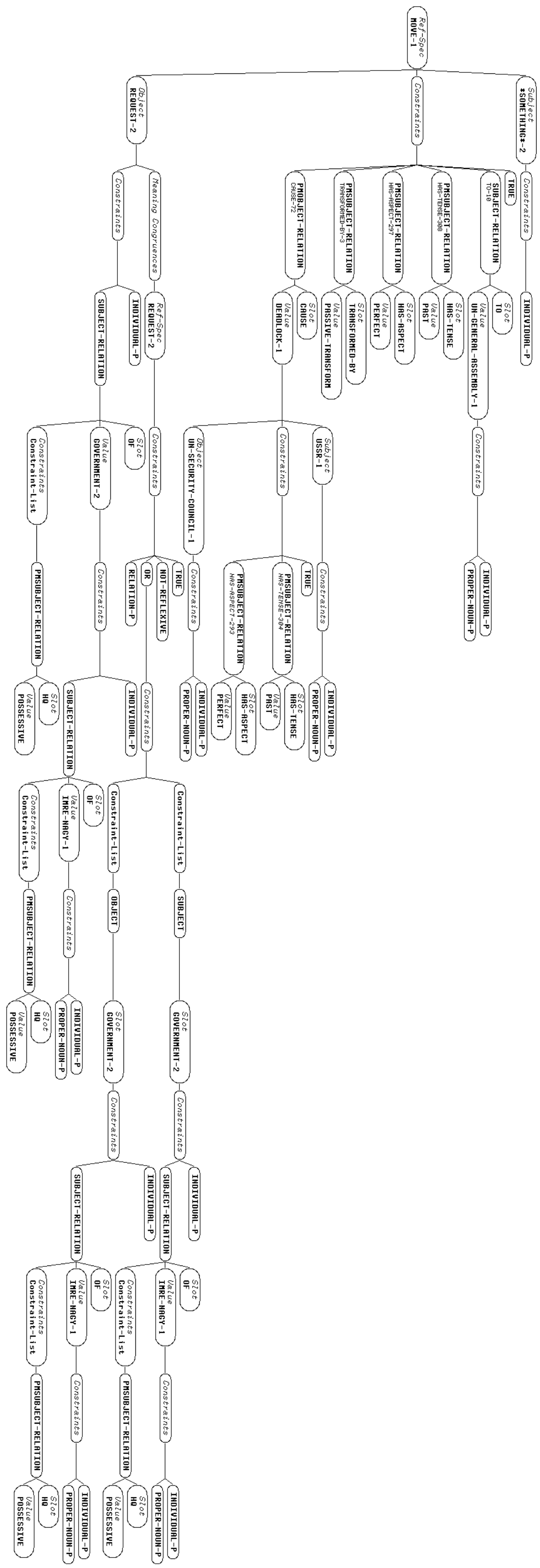
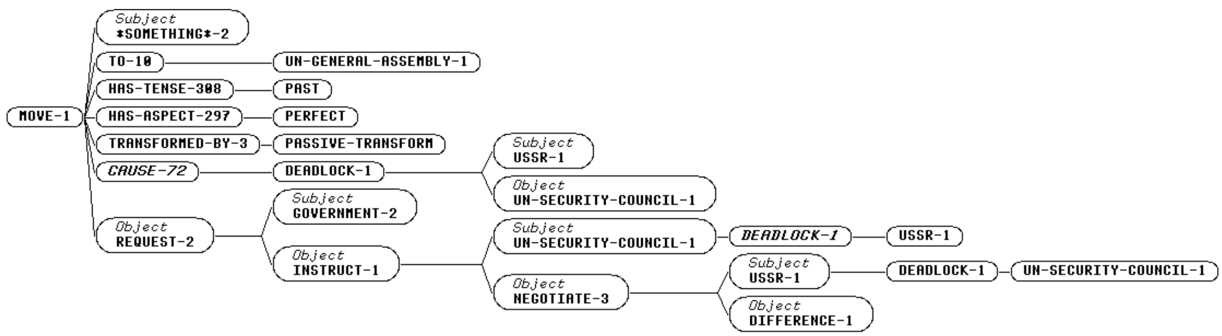


Figure 7: The semantic structure created by the syntactically-canonical sentential reference specification in figure 6. 'Request-2.' illustrates an intersentential coreference for a regular derived nominal.



2.6 Existing Text Models

Text models in several content domains from international politics have been developed using the RELATUS technology.

2.6.1 Butterworth Conflict Narratives

The first political text modeling at M.I.T. were the author's attempts model the Butterworth (1976) summary account of the 1956 Soviet Intervention in Hungary using the Winston-Katz analogy system between 1980 and 1983.[37] By 1984, the author was able to parse and fully represent in RELATUS both the Hungary story and the 1968 Soviet Intervention in Czechoslovakia. The texts each run about ten pages and create about 6000 graph structure nodes each. The coding employed Winston's (1980) conventions, making liberal use of the connective 'because' to create causal relations between clauses. These connectives allowed the Winston analogy system, and later the RELATUS analogy routines (Mallery & Hurwitz, 1987), to follow causal relations and formulate analogies. This coding was supplemented by a motivational 'for' derived from certain infinitive constructions.[38] In 1987, an undergraduate student parsed and represented several But-

[37]The text is coded in literal and explicit English from the account in (Butterworth & Scranton, 1976) between 1980 and 1983. The author originally coded the text for input to the Winston-Katz analogy system. The author successfully performed some analogical reasoning with this text, but was never able to fully represent its 5000 nodes until the advent of the representational technologies in RELATUS.

[38]An example is "Joe went to the store to get some food." This coding was jointly developed with Duffy, building from the author's analogy to the Spanish 'por' and 'para' as well the French 'pour.'

terworth summaries, including the 1954 US intervention in Guatamala, the bay of Pigs, and the Jordanian-Lebanese civil war. In 1989, Hurwitz parsed and represented the Butterworth account of the Cuban Missile Crisis for comparison to other coding schemes (Alker, Hurwitz, Mallery & Sherman, 1989).

The Butterworth's Brezinski-inspired account of the 1956 Hungarian Intervention is the source of examples for this paper. Its communitarian orientation is reflected in Butterworth's inclusion of symbolic actions in addition to the usual realist instrumental actions. The symbolic social interactions include numerous speech acts (Searle, 1969) through which parties express commitments, directives, acknowledgments, and information (Bach & Harnish, 1979). Figures 9, 10, 11 use semantic content analysis to view the Hungary story through the prism of speech acts. In international politics, Bennett (1987) characterizes nuclear deterrence between the superpowers as a social relationship whose evolution is mediated by symbolic actions, specifically the speech acts. In organization theory, Winograd and Flores (1986) understand organizations as networks of commitments, and other speech acts.[39] For social theory, Alker (1986) and Mallery (1988a: 26-27) conceptualize social systems as networks of conversations between purposeful, normatively regulated, cognitive entities embedded in specific language communities. Thus, text modeling and semantic content analysis provide a formal vehicle for studying symbolic actions.

2.6.2 Prisoner's Dilemma Protocols

Hurwitz (1990) has parsed and represented the largest amount of text (over 100 pages) yet processed by RELATUS. He has entered protocols of sequential prisoner's dilemma (SPD) games, originally acquired from a series of experiments conducted by Alker with M.I.T. undergraduates (Alker & Hurwitz, 1980). These protocols contain narrative accounts of game play augmented by natural language statements by players about their expectations, beliefs, normative responses concerning the course of play. These texts are replete with opaque contexts that must distinguished from toplevel statements.[40] An SPD analysis package finds behaviorial conflict phases in protocol representations and contains some interfaces to RELATUS specialized for SPD (Mallery, *et al.*, 1986). Hurwitz applies techniques of semantic content analysis to the text models of these protocols (Hurwitz, & Mallery, 1989; Hurwitz, 1990) His analysis shows that interactional structure in the sequential prisoner's dilemma follows from self-interpretive and reflective notions of appropriate conduct (Alker & Hurwitz, 1980; Alker & Tanaka, 1981, Alker, 1985) rather than, for example, unreflective tit-for-tat reciprocity (Axelrod, 1984).

[39]Electronic office information systems are beginning to reflect these understandings. For example, the M.I.T. Information Lens project categorizes electronic mail messages according to a simple speech act classification (Malone, *et al.*, 1987).

[40]The author developed general techniques for recognizing many cases of opacity (Mallery, *et al.*, 1986; Mallery, 1987).

2.6.3 The SHERFACS International Conflict Dataset

A feature vector editor (Mallery, 1988b) can automatically generate RELATUS text models for any cases selected 700 international conflicts or 1200 domestic disputes since 1945 in the SHERFACS International Conflict Management Dataset (Sherman, 1987a, 1987b, 1988). In April 1988, the author created a 250,000 node representation of 14 SHERFACS cases. Later in August, a lexical classification system was developed to identify Sherman conflict actions categories (Mallery, 1988c). Together, the Feature Vector Editor and RELATUS provide an environment that allows lexical classification to identify new categories in SHERFACS text models, possibly supplemented with multiple narrative accounts from varying positions, and to map them back into event data format.

2.6.4 Newspaper Articles on Vietnam

Devereux (1989) recently parsed and represented a number of articles from major US newspaper covering the September 3, 1967 presidential elections in South Vietnam. He aims to use semantic content analysis to test hypotheses of an "investment" theory of the news media. Although Devereux's analysis relies mostly on inspection of the referential integration of his text models, he also begins to devise some lexical classifiers for politically relevant concepts. He hopes to represent additional news stories and to use the collection of text models to model differential perception (section 3.4.3) with lexical classifiers (Mallery, 1988c).

2.7 Additional Facilities

2.7.1 Semantic Inversion

Since semantic representations become very complicated quickly, manual or *ad hoc* strategies for constructing reference specifications from graph structures are fragile and error prone at best. The *semantic inverter* is an interpreter that can invert the function performed by constraint-interpreting reference. In essence, semantic inversion allows one to point at some graph structure and have a reference specification constructed that will find structures like it. Given a seed node in the semantic graph structure and some constraints to delineate a graph region for incorporation, the semantic inverter traverses the graph structure by following all relations from the initial node that satisfy the set of *incorporation constraints*. As it traverses the structure, it performs some *inversion activity*, such as constructing reference specifications, which the reference system can then apply to recognize similar structures. When an application requires a constraint description that differs systematically from the traversed structure, it can provide *insertion constraints*. Insertion constraints are essentially pattern-action rules. Their pattern is either a reference specification or a LISP predicate. Their action is a LISP form that returns a set of constraints to insert at the current position in the reference specification under construction. Differing constraint descriptions can also be generalized with *variabilization levels*, numbers indicating the steps to climb in a classification hierarchy around objects or relations (see section 3.1.4). These facilities automatically tailor reference specifications for tasks such as question answering or building constraint descriptions for lexical recognizers (section 3.1.4). Another type of inversion activity generates graphical displays RELATUS knowledge structures in the *Belief Examiner Window* (section 2.7.4) while a generation activity is used by an experimental sentence generator to create surface sentences from semantic structure.[41]

2.7.2 Question Answering

The development of the first semantic inverter in 1984 was motivated by the goal for belief systems to answer various types of literal and explicit questions.[42] The key insight conceives of the graph structure created by referencing questions as a set of constraints connected to some unknown (*e.g.*, person, thing, place, truth value or enabling cause). Semantic inversion of the representation for a question, beginning from the unknown constructs a reference specification that finds the answer plus the question's unknown. Thus, the implementation adds a constraint to suppress the unknown, preventing it from appearing in the answer. The advantage of this approach for question answering is that it does not require *ad hoc* manipulations of syntactic parse graphs (Katz, 1988) for questions to produce a query specification. Instead, the normal processing for sentences is applied except that sentential referencing of the question must not resolve the reference for the unknown expressed by the question.[43] Since it exploits the normal infrastructure provided by the parser, the sentential constraint poster, and the reference system, the question answering facility requires very little mechanism beyond the semantic inverter.

The present question answering facility answers yes-no questions (including tag questions) and wh-questions (except when).[44] Questions can be asked about anything *explicitly* represented in a belief system – even things which were never stated but were combined by intersentential reference or were inferred by lexical classification. Although it is not presently implemented, backward chaining to infer implicit answers (Katz, 1988) would enhance question answering. Figure 11 shows some examples of RELATUS answering questions. In general, questions are correctly and robustly answered to the extent that the question parses and references correctly so that semantic inversion can build correct constraints that pick out the right answer. In principle, correct answers follow when texts conform to the linguistic processing model *and* the system operates without bugs modifying coverage. For this reason, question answering is an effective way to test the correct operation of the entire cycle that encodes text models.

---

[41]The sentence generator handles multiple clauses, relative clauses, as well as passive and dative transformations, but lacks numerous stylistic and morphological capabilities. The generator creates a theory-independent base structure, then uses, a theory-specific module to decode semantic structures derived from the Duffy parser. Mallery (1989) presents general principles that make a complete semantic inverter and a complete constraint interpreting reference system possible.

[42]This facility was demonstrated at the 1984 National Conference on Artificial Intelligence.

[43]Multi-sentence questions are a possible with this approach just as multiple sentences can be used to create constraint structures to for lexical classifiers (See section 3.1.4).

[44]Due to the irregularity and the popularity of copular questions, *e.g.*, "Who is Joe?", a special facility handles them.

2.7.3 Editor Mode for Text Modeling

As an extension to ZMACS, the native Emacs text editor of the Lisp Machine, the RELATUS Editor Mode features over 80 different commands useful for preparing, parsing, and representing text as well as verifying the semantic structures produced in belief systems. There are enough commands in the RELATUS mode that new users need not type any LISP forms to process their text. They can add information to the lexicon, make their own private lexicon patch files, parse text and represent it in belief systems. They can change file and buffer attributes for various aspects of the deictic context, such as the source, audience, time and belief-system. They can use various mouse-sensitive inspectors to examine the surface structure and deep structure of parses. They can view the reference specifications produced by their sentences. They can reference sentences in their chosen belief systems and examine the contents of those belief systems. These kinds of facilities are important to not just speed the task of representing text but also to make these tools available to non-programmers or anyone who is not familiar with the implementation.

2.7.4 Belief System Examiner

The *belief system examiner* provides a graphical display interface to the semantic structures of belief systems. It uses the semantic inverter to walk regions of semantic structure and generate displays. Using a display activity, the semantic inverter walks out in all directions from a "seed" node until it encounters regions that do not satisfy incorporation constraints for relations or objects. Different types of semantic structure can be displayed by tailoring incorporation constraints to restrict the graph walk to the desired structures. The examiner supports numerous display commands, including ones to show structure created by a sentence (figure 2 and figure 7), generalization hierarchies (figure 9), category instances ordered by creation time (figure 10), and causal-intention structure. For instance, all figures displaying semantic structure are created automatically by a command that scales and hardcopies any displays. It also supports commands related to the lexical classifier. It provides a convenient interface to general operations on belief systems. Since its implementation in December 1987,the belief system examiner has proved an invaluable tool for inspecting and verifying knowledge structures.

2.8 Discussion

The major effort in preparing texts for parsing is to paraphrase the original English according to the literal and explicit processing model and make it conform to the implemented grammar of English syntax. Quite often, particularly in more technical texts, sentences are fairly close to the literal and explicit model. They may not require extensive reworking. However, whenever metaphors or deliberative references occur in the sentences, they must be made explicit and literal according to the single word sense chosen for the text (or belief system) as a whole. For each sentence, one must ensure that the correct vocabulary (and lexical fields) are present, and verify the categorial disambiguations, the syntactic analyses, as well as the correctness of semantic structures. Users encounter the most difficulties obtaining correct parses, perhaps because this is the first stumbling block. Afterwards, they may complain about overly subtle distinctions in noun-phrase quantification, occasional errors in constraint posting, and more frequently, about the failure of deliberative references unsupported by the processing model. Even if this process is fairly time consuming, it only needs to be done once. Afterwards, the "debugged" text parses and references a rate of 1.2 seconds for the average sentence, which multiplies out to about 80 to 100 pages per hour.[45] A number of support tools available in the RELATUS environment, such as the editor mode, simplify and speed the text preparation process. Interestingly, the discipline of converting text to the immediate reference model force a user to think closely about what they themselves must do to understand the sentences. This reflective process is a continuing source of insights into how language works, suggesting how computers might model it. A competent RELATUS user can process about 10 pages of raw text from a new domain in a working day.[46] As domain relevant vocabulary and background knowledge are developed, the daily amount of new text processed should increase and converge to the time required to make the text meet the processing model.

3 Lexical Classification

3.1 Systems of Lexical Recognizers

3.1.1 Recognizer Organization

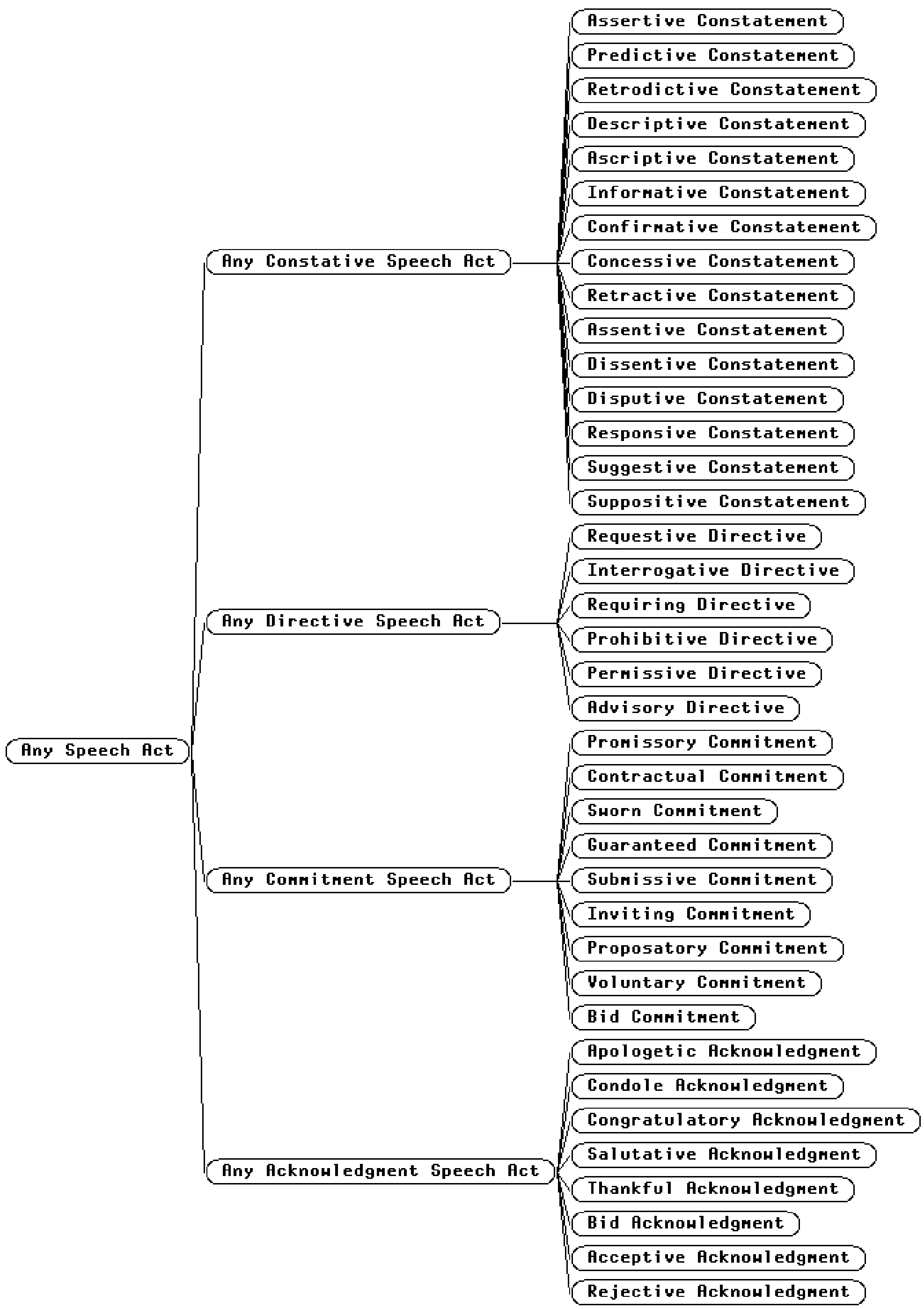
Lexical recognizers generally have associated words and selection constraints designed to find the various lexical realizations which instantiate the category. If a user defines a number of recognizers, it is convenient to organize them into independent collections, or *lexical classification systems*, and to organize the collections hierarchically (or heterarchically). When lexical recognizers are organized hierarchically, two types can be distinguished: *Base categories* generally form its leaves and recognize instances, whereas the *abstract categories*, above them in the hierarchy, have no tokens or selection constraints to recognize instances. Instead, they have specialization, generalization, and equality relationships to situate them taxonomically with respect to base categories or other abstract categories.[47] Figure 8 shows the hierarchical organization of categories for a lexical classification system that identifies speech acts (Bach & Harnish, 1979). The right-most column are base categories that recognize instances while the rest are the abstract categories that link them into a taxonomic hierarchy.

[45]These timings are for an older and slower Symbolics 3650 Lisp Machine with four megawords of real memory and a 360 megabyte CDC disk drive.

[46]Attaining this level of competence may require several months of study. Patrick Winston reports that M.I.T. undergraduates have considerably difficulty building text models because they expect performance beyond the processing model implemented by the Winston-Katz system (personal communication, September, 1989).

[47]The implementation allows base categories to have specializations.

Figure 8: The hierarchical organization of categories in a lexical classification system for the Bach and Harnish (1979) speech act taxonomy. The right-most column are base categories that recognize instances. The rest are the abstract categories that link them taxonomically.



3.1.2 Instance Classification

Although some commands in the RELATUS text editor mode and the belief system examiner simply find instances of categories and view their source sentences, as seen in figure 11, it is generally more useful to label the category instances. For this reason, every lexical recognizer should have an associated *concept reference specification*. This makes possible commands in the belief system examiner and editor to lexically classify instances. Figure 9 shows the partial instantiation of the Bach and Harnish (1979) taxonomy for speech acts found in the Hungary story. The taxonomic connective is the italicized object relation 'be.' The column headed by 'constatement-8' contains the semantic representation of base lexical recognizers, which their concept ref-specs create. To the left, taxonomic links connect these nodes to the semantic representations of abstract categories. To the right, taxonomic links connect the base categories to their instances, the speech acts from the text. Lexical classification makes available the categorizations of a text model for other uses. These include displaying concept instances (figure 10) or answering questions (sentence-271 in figure 12).

3.1.3 Inspecting Results

A user can inspect the results of lexical classification using the standard tools for examining semantic structure. The belief system examiner (section 2.7.4) has a number of commands and display modes to view lexical classifiers (figure 8), their constraints, and the structures they pick out in text models (figures 9, 10). It is also possible to retrieve the source sentences for semantic structures to see if classifications conform to usages in the source text (figure 11). The RELATUS editor mode (section 2.7.3) supports many of the same commands but does not use graphical displays for presentation. Both the belief system examiner and the editor allow a user to browse through semantic structure using a mouse-sensitive "frame" inspector. Although question answering based on classifications can provide another means of inspecting results (figure 12), people can more effectively look at the knowledge representation than formulate questions to test classification correctness. In general, all the implemented inspection methods rely on a human who examines, either directly or indirectly, the classifications and the reasons for the classifications.

3.1.4 Defining Lexical Classifiers

Earlier versions of the RELATUS lexical classification system (Mallery, 1987, 1988b, 1988c) required the analyst to define categories by invoking a LISP definition form and supplying constraint specifications (see figure 16). Classifier definition required a rudimentary knowledge of LISP and some familiarity with the constraint language. In general, the ability of users to write pattern specifications limited the complexity of lexical classifiers and recognitions. The recent introduction of an editing interface for lexical classifiers (figures 13, 14, 15) pushed back these limits. The editing interface

- automatically writes LISP definitions for lexical classifiers based on information elicited from users via an advanced window interface;

Figure 9: The taxonomy of speech-acts found in the text model for the 1956 Hungarian intervention. The column headed by 'constatement-1' contains the semantic representation of base lexical recognizers. To the left, taxonomic connective (the italicized object relation 'be') links these nodes to abstract categories. To the right, taxonomic connective links base categories to the speech acts in the text model.

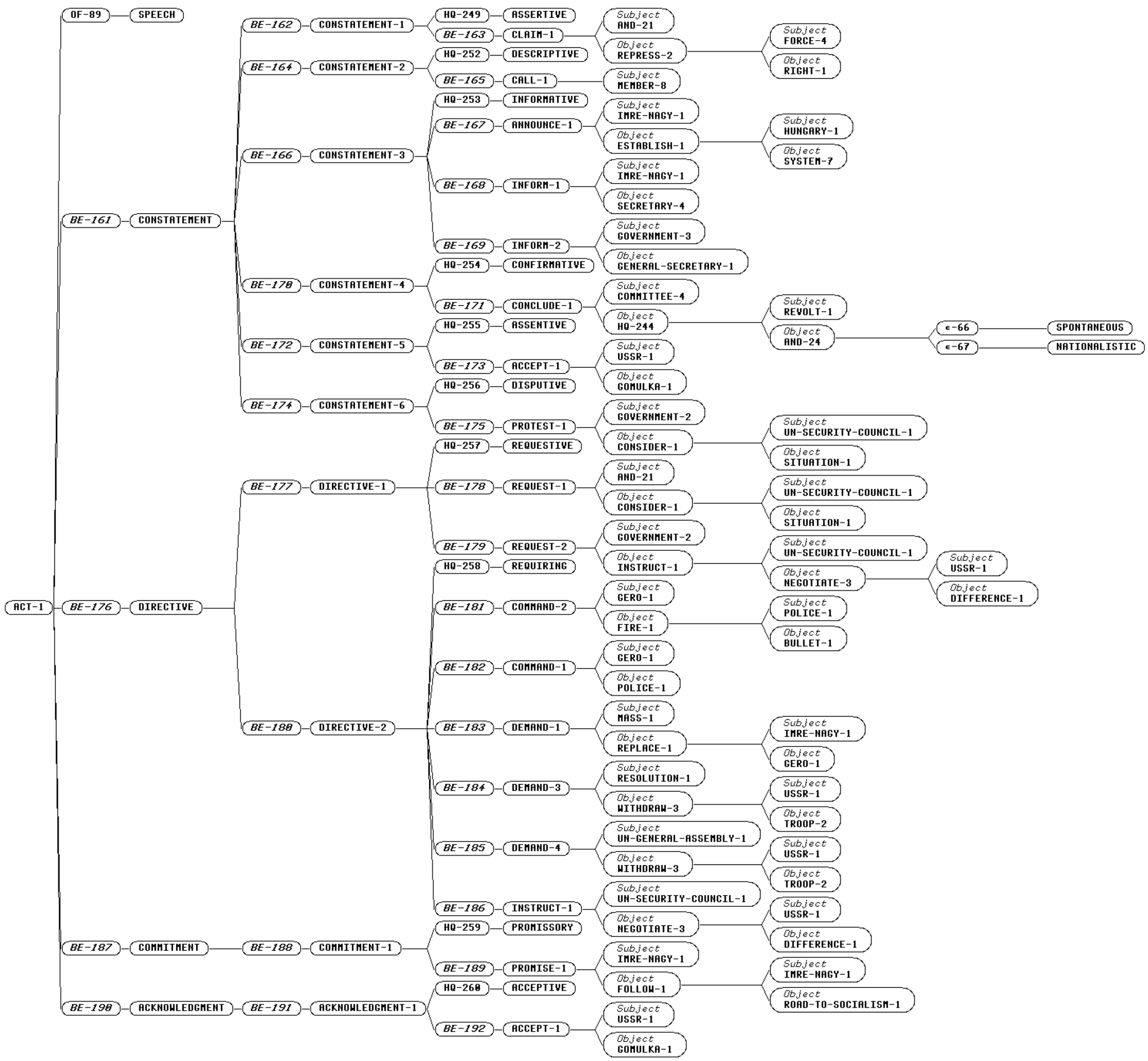


Figure 10: The semantic structures for each speech act in the 1956 Hungarian intervention. Lexical classification allows presentation of sequences of instances to summarize the text model from the perspective of the category.

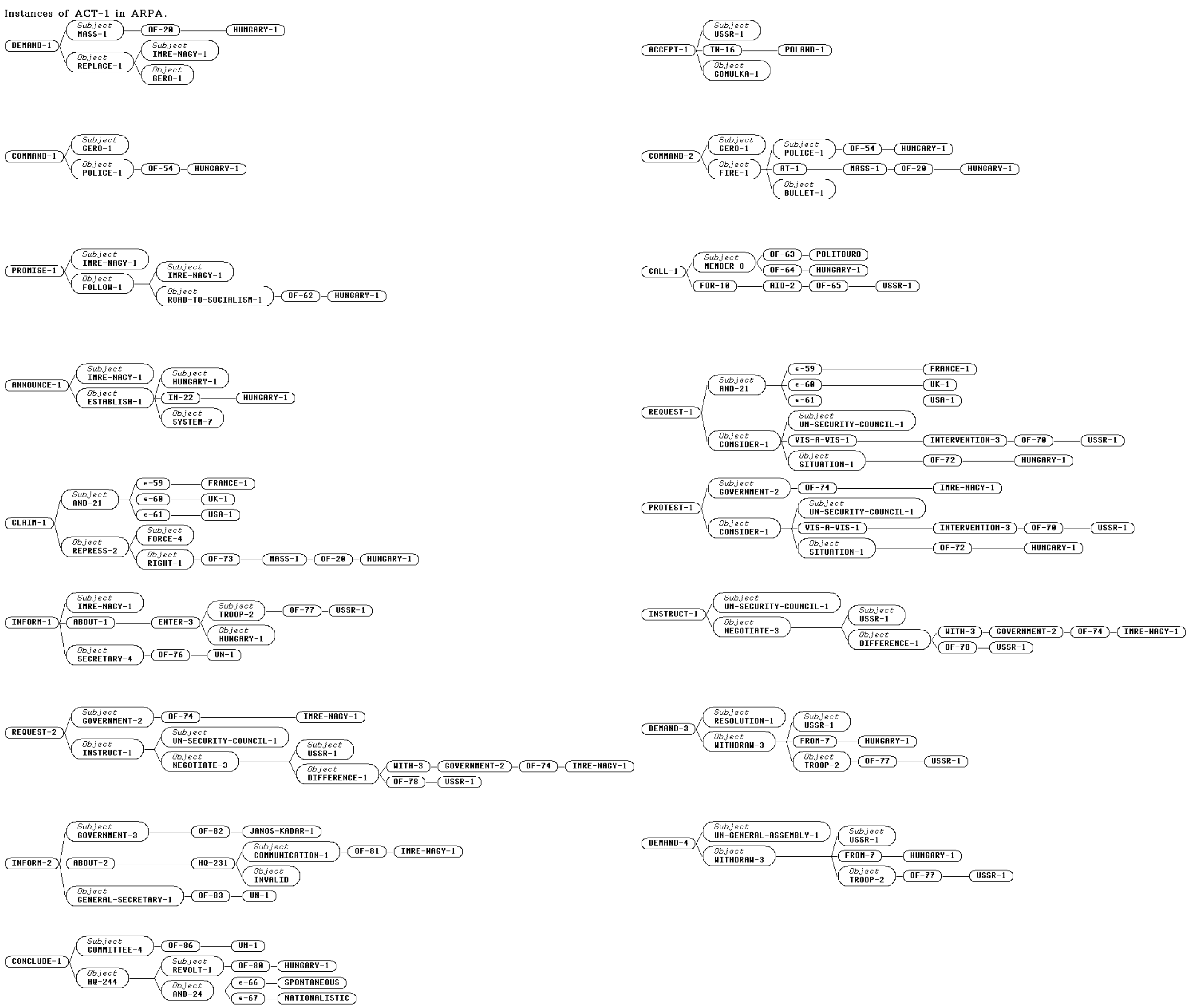


Figure 11: The speech-act sentences in the 1956 Hungarian intervention. The gennamed verbs denote the semantic node referred to in the accompanying sentence. Multiple appearances of the same genname reflect intersentential references.

SPEECH-ACT (ACT-1) sentences in ARPA:

DEMAND-1: The masses demanded that Imre-Nagy replace Gero because the masses were politicized and because Gero did not satisfy the masses and because the masses believed that the USSR had legitimated national Communism.

ACCEPT-1: The masses believed that the USSR had legitimated national Communism because the USSR accepted Gomulka in Poland.

COMMAND-1: Gero was able to command the secret police because the secret police supported him.

COMMAND-2: The secret police fired some bullets at the masses because the masses demonstrated for Imre-Nagy and because Gero commanded the secret police to fire the bullets at the masses.

COMMAND-2: Gero commanded the secret police to fire the bullets at the masses because Gero feared that he would fall from political power.

PROMISE-1: Imre-Nagy promised to follow a Hungarian road-to-socialism.

CALL-1: A Hungarian politburo member called for direct Soviet military aid.

ANNOUNCE-1: Imre-Nagy announced that Hungary would establish a multi-party political system in Hungary.

REQUEST-1: France, the UK, and the USA requested that the UN-Security-Council consider Hungary's situation vis-a-vis the Soviet intervention.

CLAIM-1: France, the UK, and the USA claimed that foreign military forces were repressing the masses' rights.

PROTEST-1: Imre-Nagy's government protested the UN-Security-Council's consideration because the situation was within Hungary's domestic jurisdiction.

INFORM-1: Imre-Nagy informed the UN's general secretary that additional Soviet troops had entered Hungary.

INSTRUCT-1: Imre-Nagy's government requested that the UN-Security-Council instruct the USSR to negotiate its differences with Imre-Nagy's government.

REQUEST-2: Imre-Nagy's government requested that the UN-Security-Council instruct the USSR to negotiate its differences with Imre-Nagy's government.

REQUEST-2: The USSR deadlocked the UN-Security-Council because the USSR did not want the UN-Security-Council to consider Imre-Nagy's government's request and because the USSR was able to deadlock the UN-Security-Council.

REQUEST-2: The UN-Security-Council did not consider Imre-Nagy's government's request because the USSR deadlocked the UN-Security-Council.

REQUEST-2: Imre-Nagy's government's request was moved to the UN-General-Assembly because the USSR deadlocked the UN-Security-Council.

DEMAND-3: The UN-General-Assembly adopted a resolution which demanded that the USSR immediately withdraw the soviet troops from Hungary.

INFORM-2: Janos-Kadar's government informed the UN General-Secretary that Imre-Nagy's communications were invalid.

DEMAND-4: The UN-General-Assembly continued to demand that the USSR withdraw its troops from Hungary although it was clear that Soviet policy would not be deflected and that the Soviet intervention had put down the Hungarian revolt.

CONCLUDE-1: The special committee concluded that the revolt had been spontaneous and nationalistic.

Figure 12: Answering questions about the 1956 Hungarian intervention. Questions answering provides a means to inspect the text model and to ascertain lexical classifications.

```
Question to ask ARPA: (<END> to end)
> Why did Imre-Nagy fall from political power?

ARPA: FALL-4's causes are:  PUT-1

Question to ask ARPA: (<END> to end)
> Who was executed?

ARPA: For the question, EXECUTE-3, I find: IMRE-NAGY-1 ASSOCIATE-3

Question to ask ARPA: (<END> to end)
> Did the UN-General-Assembly demand that the USSR withdraw its troops from Hungary?

ARPA: Yes, because DEMAND-4 is TRUE.

Question to ask ARPA: (<END> to end)
> Did the USSR withdraw its troops from Hungary?

ARPA: Since WITHDRAW-3 is within an opaque belief context, I cannot be sure.

Question to ask ARPA: (<END> to end)
> What directives are there?

ARPA: There are:  REQUEST-1 REQUEST-2 COMMAND-2 COMMAND-1 DEMAND-1 DEMAND-3 DEMAND-4 INSTRUCT-1

Question to ask ARPA: (<END> to end)
> What retractive constatements are there?

ARPA: There are no CONSTATEMENT-7s.

Question to ask ARPA: (<END> to end)
> What perlocutionary relations are there?

ARPA: There are:  CAUSE-45
```

- automatically creates pattern descriptions in the constraint language based on a series of English sentences provided by users.

The definition of pattern recognizers with English sentences achieves numerous beneficial goals.

- **Less Manual Writing of Constraint Specifications:** Users need little knowledge of the constraint language for the reference system.
- **Less Knowledge of Logical Form:** Users need not know the specific methods for encoding the semantic content of English sentences.
- **User Friendliness:** The system becomes available to less computer literate people.
- **More Complex Recognition Tasks:** A series of sentences more easily specifies complex patterns than hand-coding in the constraint language.
- **Error Reduction:** Specified patterns use the constraint language without syntactic errors that might require tedious debugging.
- **Greater Productivity:** Patterns are more quickly specified, modified, and tested with an automated acquisition interface.
- **Greater Task Focus:** Freedom from details of hand-coding lexical classifiers allows users to concentrate on the substantive recognition tasks.

The interface uses the Symbolics Lisp Machine Presentation System (Symbolics, 1988) to accept the correct range of values for the various parameters that make up a lexical classifier.[48] The limitations on the range of values that users can supply reduce the possibility of error, making the man-machine process more reliable. Figure 13 shows an example of editing a lexical classifier for a speech act.

Since semantic categories may be expressed in texts by different lexical realizations or paraphrasing, a lexical classifier for the category needs to look for each alternative. Thus, the interface prompts for constraint descriptions for any number of lexical realizations. There are three phases to specify each a lexical realization's constraint description.

- **Specification sentences:** The user provides sentences to characterize it (figure 14).
- **Specification Representation:** The sentences are syntactically parsed and semantically represented to yield a referentially-integrated graph structure.
- **Constraint Description Construction:** The semantic inverter (see section 2.7.1) traverses the associated semantic representation and creates a constraint description to recognize *similar* knowledge structures (figure 15).

[48] Recent meetings on standards for the Common LISP Window System have decided to adopt a window system and presentation standard similar to the Symbolics ones. Within several years this technology will be generally available to the rest of the Common LISP community.

Figure 13: Editing a lexical classifier for retractive speech acts. The user is adding the concept ref-spec by providing an English noun phrase for conversion into a constraint description (shown in figure 14).

```
:Edit Lexical Classifier (New classifier [default No]) No
Lexical category: Retractive-Constatement

Editing Lexical Classifier

Lexical Category: RETRACTIVE-CONSTATEMENT
Documentation: Finds retractive constatements.
Graph Category: Relation  Object  Token-Universal  Universal  Individual  Verb  Adjective  Adverb
Generalizations: Constatement
Specializations: keywords or a null value
Equivalences: keywords or a null value
Opaque category: Yes  No
Lexical Type: Retractive Constatement
Lexical Classification System: Bach & Harnish Speech Acts
Concept Ref-Spec: (Noun Phrase) a retractive constatement
Lexical Realization 1: (ABJURE)
Lexical Realization 2: (CORRECT)
Lexical Realization 3: (DENY)
Lexical Realization 4: (DISAVOW)
Lexical Realization 5: (DISCLAIM)
Lexical Realization 6: (DISOWN)
Lexical Realization 7: (RECANT)
Lexical Realization 8: (RENOUNCE)
Lexical Realization 9: (REPUDIATE)
Lexical Realization 10: (RETRACT)
Lexical Realization 11: (TAKE ((((SUBJECT-RELATION HAS-PARTICLE BACK ((TRUE)))
                                 (OBJECT-CONSTRAINTS
                                   ((OR (((RELATION-P))
                                         ((CLASS (REF-SPEC PROPOSITION
                                                   :QUANTIFICATION :TOKEN-UNIVERSAL))))))))))
Lexical Realization 12: a lexical realization
<ABORT> aborts, <END> uses these values
```

The acquisition interface presents alternate lexical realizations as a list of the token and the selection constraints that distinguish appearances of the token fitting the category.[49] Users must provide the word stem and some sentences (figure 14). Normally, one sentence will use the token in an example of the lexical realization. Subsequent sentences may provide additional structure, such as classificational information about other semantic objects appearing in the original sentence.[50]

Figure 14: Providing English sentences for conversion into the constraints for realization 12 (shown in figure 15). These constraints will allow the lexical classifier to find and disambiguate the retractive sense of the verb 'withdraw.'

```
Editing Lexical Classifier

Lexical Category: RETRACTIVE-CONSTATEMENT
Documentation: Finds retractive constatements.
Graph Category: Relation  Object  Token-Universal  Universal  Individual  Verb  Adjective  Adverb
Generalizations: Constatement
Specializations: keywords or a null value
Equivalences: keywords or a null value
Opaque category: Yes  No
Lexical Type: Retractive Constatement
Lexical Classification System: Bach & Harnish Speech Acts
Concept Ref-Spec: (REF-SPEC CONSTATEMENT
                            :CONSTRAINTS ((MINIMAL-INITIAL-DESCRIPTION
                                            ((UNIVERSAL-P)
                                             (SUBJECT-RELATION HQ RETRACTIVE ((TRUE)))
                                             (PNUMBER-OF-SUBJECT-RELATIONS HQ 1))))
                            :MODE :REFERENCE)
Lexical Realization 1: (ABJURE)
Lexical Realization 2: (CORRECT)
Lexical Realization 3: (DENY)
Lexical Realization 4: (DISAVOW)
Lexical Realization 5: (DISCLAIM)
Lexical Realization 6: (DISOWN)
Lexical Realization 7: (RECANT)
Lexical Realization 8: (RENOUNCE)
Lexical Realization 9: (REPUDIATE)
Lexical Realization 10: (RETRACT)
Lexical Realization 11: (TAKE ((((SUBJECT-RELATION HAS-PARTICLE BACK ((TRUE)))
                                 (OBJECT-CONSTRAINTS
                                   ((OR (((RELATION-P))
                                         ((CLASS (REF-SPEC PROPOSITION
                                                  :QUANTIFICATION :TOKEN-UNIVERSAL))))))))))
Lexical Realization 12:
(Lexical token) WITHDRAW
(Sentences) A person withdraws the statement. The statement is a proposition.
(Object variablization level) 1
(Token constraints) a constraint structure
<ABORT> aborts, <END> uses these values
```

[49]Categories are typically word senses; but they need not be. They could be more fine grained or they might recognize structures not normally associated with specific words.

[50]If the token appears more than once in the sentences *and* it has different referents in the different appearances, users need to supply some constraints for the *token-constraints* value that the system will use to find the intended token.

Finally, users can control the application of a simple induction heuristic,[51] the *object variabilization level.*

- When set to 0, the semantic inverter returns a constraint description of the relational embedding of the node, which includes the classification[52] for each object and relation in the embedding.
- When set to 1, the semantic inverter converts every object (except the root token) to variables by dropping the restriction on the token type of matched nodes while retaining just relational embedding and classification.
- When set above 1, the level is interpreted as the number of steps to climb in the class hierarchy spanning an object to obtain its classification. Thus, the class constraint will incorporate the first level classes for an object variabilization of 1 (most specific classification) or will incorporate the second level classes for 2.

Object variabilization uses the "climb class hierachy" induction heuristic to generalize recognition constraints from single examples, a simple form of "explanation-based learning," (Mitchell, *et al.*, 1986).[53] For now, users must circumscribe the content of specification sentences, or must manually edit the resulting constraint descriptions, to operationalize other induction heuristics, such as "drop-link."

If the category is to label instances, users must supply the concept reference specification. After users provides an English noun phrase to parse and reference in the knowledge representation (see figure 13), the system semantically inverts the represented structure and produces the concept ref-spec of the lexical classifier (see figure 14).

In the typical editing cycle, a user defines the newly edited lexical classifier, and then, tests it, using invocation interfaces in the Editor Mode or the Belief System Examiner. If the lexical classifier retrieves instances which are not members of the category (incorrectness) or fails to retrieve instances of the category (incompleteness), the user may edit the lexical classifier to add or remove constraining information. Once recognitions are adequate, the individual definitions can be collected in LISP files that define a lexical classification system. Thereafter, reloading the compiled versions of these files reinstantiates the entire lexical classification system.

3.1.5 Hierarchical Classification

Hierarchical lexical classification uses prior classifications in subsequent ones. For example, after recognizing speech acts, another lexical classifier (figure 17) could find perlocutionary force (effects on others) of speech acts. Figure 18 illustrates one such recognition, where the Soviet acceptance ('accept-1') of Gomulka in Poland caused the

[51]Michalski (1987) overviews induction methods.

[52]A node's classification includes its classes and those classes to which it is known not to belong, its disjoint classes.

[53]Although the semantic inverter supports relation variabilization, it has not yet been incorporated into this interface yet.

Figure 15: The edited lexical classifier for retractive speech acts. The classifier now has a constraint description in lexical realization 12 to pick out the sense of 'withdraw' as a speech act and a concept ref-spec to label instances.

```
Editing Lexical Classifier

Lexical Category: RETRACTIVE-CONSTATEMENT
Documentation: Finds retractive constatements.
Graph Category: Relation  Object  Token-Universal  Universal  Individual  Verb  Adjective  Adverb
Generalizations: Constatement
Specializations: keywords or a null value
Equivalences: keywords or a null value
Opaque category: Yes  No
Lexical Type: Retractive Constatement
Lexical Classification System: Bach & Harnish Speech Acts
Concept Ref-Spec: (REF-SPEC CONSTATEMENT
                            :CONSTRAINTS ((MINIMAL-INITIAL-DESCRIPTION
                                            ((UNIVERSAL-P)
                                             (SUBJECT-RELATION HQ RETRACTIVE ((TRUE)))
                                             (PNUMBER-OF-SUBJECT-RELATIONS HQ 1))))
                            :MODE :REFERENCE)
Lexical Realization 1: (ABJURE)
Lexical Realization 2: (CORRECT)
Lexical Realization 3: (DENY)
Lexical Realization 4: (DISAVOW)
Lexical Realization 5: (DISCLAIM)
Lexical Realization 6: (DISOWN)
Lexical Realization 7: (RECANT)
Lexical Realization 8: (RENOUNCE)
Lexical Realization 9: (REPUDIATE)
Lexical Realization 10: (RETRACT)
Lexical Realization 11: (TAKE ((((SUBJECT-RELATION HAS-PARTICLE BACK ((TRUE)))
                                  (OBJECT-CONSTRAINTS
                                    ((OR (((RELATION-P))
                                          ((CLASS (REF-SPEC PROPOSITION
                                                            :QUANTIFICATION :TOKEN-UNIVERSAL))))))))))
Lexical Realization 12: (WITHDRAW (((OBJECT-CONSTRAINTS
                                      ((OR (((RELATION-P))
                                            ((CLASS ((REF-SPEC PROPOSITION
                                                               :QUANTIFICATION :TOKEN-UNIVERSAL
                                                               :MODE :REFERENCE))))))
                                       (RELATION-P))))))
Lexical Realization 13: a lexical realization
<ABORT> aborts, <END> uses these values
```

Figure 16: The Lisp definition of the lexical classifier for retractive speech acts created using the definition interface shown in figures 13, 14 and 15. Before the advent of the interactive editing interface, the user would have specified this definition directly in Lisp.

```
(DEFINE-LEXICAL-CATEGORY
  RETRACTIVE-CONSTATEMENT
  "Finds retractive constatements."
  :VERB
  :GENERALIZATIONS (:CONSTATEMENT)
  :LEXICAL-TOKENS
  (ABJURE CORRECT DENY DISAVOW DISCLAIM DISOWN RECANT RENOUNCE REPUDIATE RETRACT TAKE WITHDRAW)
  :SELECTION-CONSTRAINTS
  ((TAKE
     (((SUBJECT-RELATION HAS-PARTICLE BACK ((TRUE)))
       (OBJECT-CONSTRAINTS
         ((OR (((RELATION-P))
               ((CLASS (REF-SPEC PROPOSITION
                                 :QUANTIFICATION :TOKEN-UNIVERSAL)))))))))
   (WITHDRAW
     ((OBJECT-CONSTRAINTS
        ((OR (((RELATION-P))
              ((CLASS ((REF-SPEC PROPOSITION
                                 :QUANTIFICATION :TOKEN-UNIVERSAL
                                 :MODE :REFERENCE)))))
         (RELATION-P))))))
  :CONCEPT-REF-SPEC
  (REF-SPEC CONSTATEMENT
            :CONSTRAINTS ((MINIMAL-INITIAL-DESCRIPTION
                           ((UNIVERSAL-P)
                            (SUBJECT-RELATION HQ RETRACTIVE ((TRUE)))
                            (PNUMBER-OF-SUBJECT-RELATIONS HQ 1))))
            :MODE :REFERENCE)
  :OPAQUE-P T
  :LEXICAL-TYPE-STRING "Retractive Constatement"
  :LEXICAL-CLASSIFICATION-SYSTEM "Bach & Harnish Speech Acts")
```

Hungarian masses to believe ('believe-1') it had legitimated ('legitimate-1') national communism. Figure 18 also shows that causal links continue from the masses beliefs to their demands ('demand-1') for Imre Nagy to replace ('replace-1') Gero – reinforced by their politicization ('politicize-1')and their dissatisfaction ('satisfy-1') with Gero. Instead of incorporating speech acts into every lexical recognizer that needs to test if a relation is a speech act, abstraction and modularity are best served by maintaining separate classification systems, simply running them in the order of their dependence, if unidirectional, or repeatedly to quiescence, if interdependent.

Although tractable in small applications, larger applications cannot afford the overhead of irrelevant checks for category instances unnecessary for the hierarchical recognition. *Hierarchical lexical classifiers* address this problem (Mallery, 1988c). They are just like ordinary base categories except that they use classifications by other lexical recognizers in their recognition constraints. The implementation records the dependencies on prior recognitions and ensures those recognizers run first. The easy way to think of hierarchical recognizers is to consider the base categories the first layer of recognizers and the hierarchical recognizers as subsequent layers. The importance of hierarchical recognizers is that they allow composition of complex patterns from smaller patterns, which each recognize some coherent part. There are benefits; component categories are recognized along with composite categories and recognizers for composite categories are easier to debug. Thus, a complex frame or script could be implemented bottom-up with hierarchical recognizers that detect slots or roles, and finally, a top-level recognizer that recognizes its instantiation in the pattern of recognized slots.

3.2 Bootstrapping Reference

Figure 16 shows a lexical recognizer that illustrates within-part-of-speech sense disambiguation for the verbs 'take' and 'withdraw'.[54] Since the *retractive* category in the Bach and Harnish speech act taxonomy is intended to recognize retractions of propositions, selection constraints appear for 'withdraw' and 'take.'[55] The sense of 'withdraw' is the meaning in which the verb takes a proposition as its object. The selection constraints for 'withdraw' in figure 16 require the object of the 'withdraw' relation to be either a relation or a proposition, which effectively discriminates the sense of 'withdraw,' as in "withdrawing troops."[56] A similar idea stands behind the selection constraints for 'take.'[57]

[54]Since the Duffy parser performs part of speech disambiguation during syntactic analysis, lexical classification needs only to cope with word disambiguation within each part of speech. The constraint poster encodes each part of speech in semantically unambiguously ways that lexical classifiers may exploit.

[55]The addition of selection constraints to the Bach and Harnish speech act taxonomy was driven by failed classifications. Consequently, selection constraints appear for only words that presented problems in our texts.

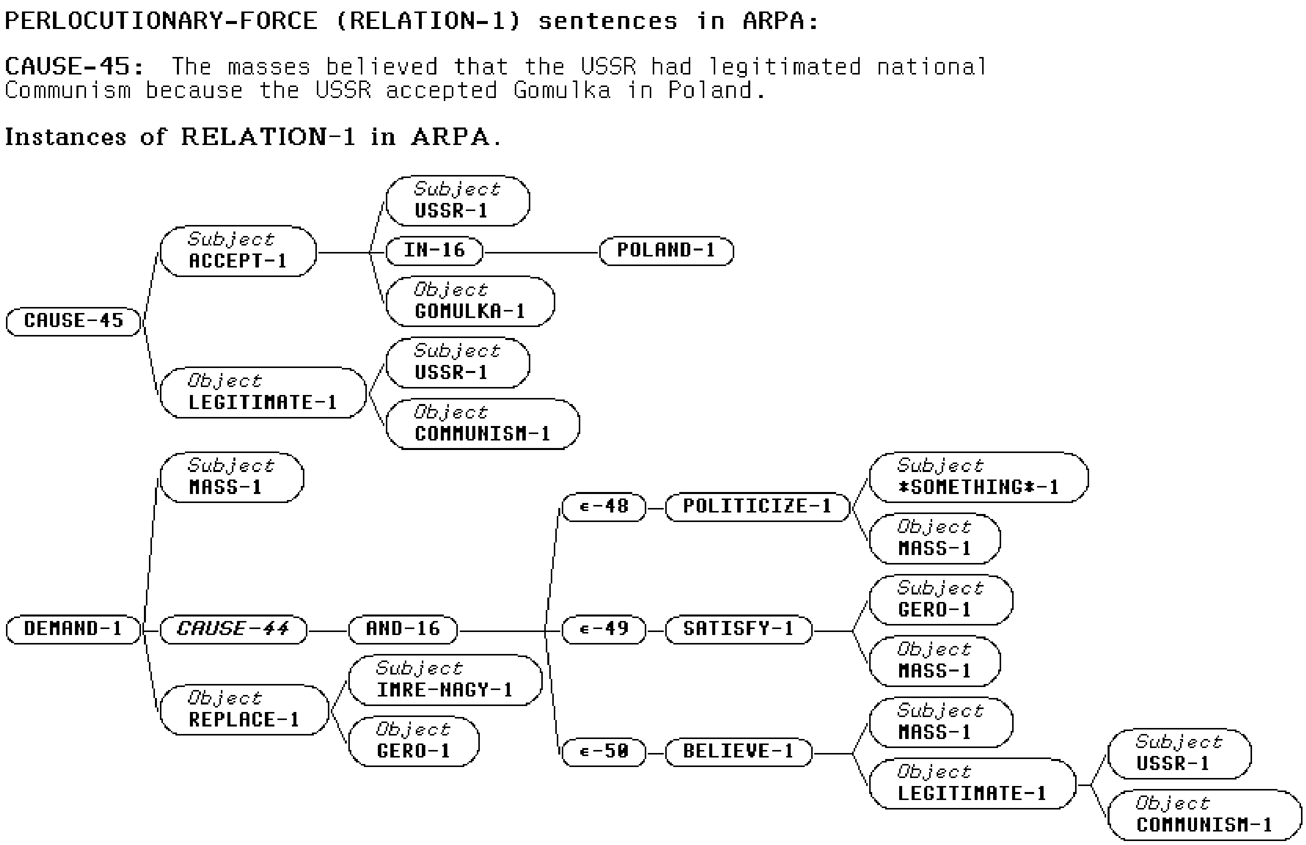
[56]Figures 14 and 15 show how the selection constraints for withdraw were provided to the system. The constraints that resulted from the specification sentences in figure 14 were manually edited to eliminate superfluous constraint structure and achieve better performance. The relation constraint was added in order to catch speech-act's whose objects are not, but should be, classified as propositions.

[57]These examples merely illustrate the mechanism and are not intended to be definitive.

Figure 17: The Lisp definition for a hierarchical lexical classifier that finds the perlocutionary force of speech acts. It looks for 'cause' relations and motivational 'for' relations whose subject is a speech act. It depends on the prior lexical classification of speech acts.

```
(DEFINE-LEXICAL-CATEGORY
  PERLOCUTIONARY-FORCE
  "Finds relations caused by speech acts."
  :RELATION
  :LEXICAL-TOKENS (CAUSE FOR)
  :CONCEPT-REF-SPEC
  (REF-SPEC
    RELATION
    :CONSTRAINTS
    ((MINIMAL-INITIAL-DESCRIPTION
       ((UNIVERSAL-P)
        (SUBJECT-RELATION HQ PERLOCUTIONARY ((TRUE)))
        (PNUMBER-OF-SUBJECT-RELATIONS HQ 1))))
    :MODE
    :REFERENCE)
  :SELECTION-CONSTRAINTS
  ((* ((SUBJECT-CONSTRAINTS
        ((RELATION-P)
         (CLASS ((REF-SPEC
                   ACT
                   :CONSTRAINTS
                   ((MINIMAL-INITIAL-DESCRIPTION
                      ((SUBJECT-RELATION OF
                                         (REF-SPEC SPEECH
                                                   :QUANTIFICATION :TOKEN-UNIVERSAL
                                                   :MODE :INTERLEAVE)
                                         ((PMSUBJECT-RELATION HQ TIGHTLY-BOUND))))))
                   :QUANTIFICATION :UNIVERSAL
                   :MODE :INTERLEAVE)))))
       (OBJECT-CONSTRAINTS ((RELATION-P)))
       (RELATION-P))))
  :LEXICAL-TYPE-STRING "Perlocutionary Force"
  :LEXICAL-CLASSIFICATION-SYSTEM "Standard Lexical Classifications")
```

Figure 18: The source sentence for a perlocutionary act and the semantic structure lexically classified as the perlocutionary force in the 1956 Soviet intervention in Hungary. The second graph shows the causal structure following from the masses belief (believe-1) that the Soviet Union had legitimated national communism to their demand (demand-1) for Gero's replacement by Imre Nagy.



Word sense disambiguation divides into deliberative and non-deliberative disambiguations. Often, words can be disambiguated by the class of words they are related to in the grammatical context of the sentence. Other times, more sentences referring to the words must supply missing category information or additional grammatical constraints. But sometimes, the only means of disambiguating words senses involves, perhaps, arbitrarily difficult reasoning. In the first two cases, lexical classification together with syntactic analysis is enough. The last case requires the help of a reasoning system. Even then, ambiguities may remain because the text is truly ambiguous. The research task for sense disambiguation is to develop a lexicon of reliable disambiguation constraints for words and to identify the class membership of words. An open question concerns how much coverage such bottom-up methods can achieve without resort to deliberation.

The term selection constraints is analogous to Katz and Fodors' (1963) notion of "selection restrictions." The difference is that Katz and Fodor grounded their theory of sense disambiguation in a decompositional semantics. A different term was needed for the application of lexical classification to the disambiguation task because the operations are taking place in a lexicalist and referentially integrated semantic representation. Moreover, rather than factoring propositions into some set of primitive classes, the discriminating referential constraints are semantic relationships.

If a natural language system classifies all new input, it increases the information explicitly encoded in the semantic representation, which becomes available as new indices. Thus, references involving classificational restrictions can succeed. These cases include relative clauses (who, which) involving class attributions, adjectives imparting category restrictions, or modified demonstrative pronouns. By extending indexation, lexical classification on input may allow the reference system to access smaller initial possibility spaces, speeding reference for these cases.

The preceding discussion has pointed to the role of lexical classification in deliberative reference (see section 2.2). Deliberative reference presents many difficulties because nothing is known about how, or if, different inference strategies are selected. People may pursue multiple inference strategies in parallel. Also, there are probably significant interactions between the lexical classification, sense disambiguation, and reference that further complicate the picture.

3.3 Existing Lexical Classification Systems

To date, there are six lexical classification systems.

- **Lasswell Value Dictionary:** The Lasswell Value Dictionary (Stone, *et al.*, 1966; Lasswell & Namenwirth, 1969; Namenwirth & Weber, 1987) was converted into a lexical classification system in October, 1987. The categories were organized taxonomically, but they are too extensive for inclusion here. Practitioners of traditional computerized content analysis expended considerable effort to develop this 8000-word dictionary and to verify the stability of its categories.[58]

[58]Because the string-oriented disambiguation rules of the Lasswell Value Dictionary have not been converted to RELATUS-style, grammatically-based constraints, the fruit of this earlier work have not

- **General Problem-Solving:** A general lexical classification system for belief, intention, and affect words has been developed primarily as an example, but was used to examine strategic language, including a political actor's model of an adversary's (or an ally's) model of themselves (Mallery, 1987).

- **Sequential Prisoner's Dilemma:** Roger Hurwitz developed a lexical classification system for SPD protocols that identifies categories associated with conflict and social order formation (Hurwitz & Mallery, 1989; Hurwitz, 1990).

- **Bach & Harnish Speech Acts:** A lexical classification system has been created for the Bach and Harnish (1979) taxonomy of speech acts (figure 8). Although the authors did not provide sense-disambiguation information, the addition of some selection constraints partially corrects this drawback. This classification system provides interesting insights for the Butterworth conflict narratives, particularly the Hungary story (figures 9, 10, 11).[59]

- **Ortony Affect Lexicon:** A lexical classification system has been created for the Ortony Affective Lexicon (Ortony, *et al.*, 1987). It distinguishes various affect classes, such as internal versus external affect and behavioral versus cognitive affect. This system also suffers from the absence of selectional constraints.

- **SHERFACS Actions:** A lexical classification system was defined for application to narrative precis automatically generated (Mallery, 1988b) from the SHERFACS International Conflict Management Dataset (Sherman, 1987a, 1987b, 1988). This system categorizes the various actions that a political actor may take in conflicts as coded in SHERFACS. These action categories include the COPDAB categories (Azar, 1982).[60]

3.4 Analytical Applications

3.4.1 Semantic Content Analysis

The immediate political-analytic application of lexical classification is semantic content analysis. The tools presented above open a universe of ways to analyze texts, leaving behind many problems of traditional computerized content analysis, but bringing some new ones. This section anticipates some evaluational issues for the methodology.

Traditional content analysis has already faced the issues of reliability and validity.[61] It evaluates results by considering the reliability of a model and the validity of the com-

yet been fully gleaned. Just how much can be recovered remains uncertain. The absence of an explicit idea of syntax in the disambiguation rules may require manual conversion to RELATUS constraints. Several difficiencies may limit the utility of this lexicon: it assumes a universal set of categories and lexical predictors; it reifies the linguistic biases of generations of Harvard undergraduates who coded it.

[59] The lexicon appears as an appendix to (Mallery, 1987).

[60] Mallery (1988c) reports on these classifiers.

[61] This section draws on Weber (1985: 16-21) and Krippendorf (1980: 130-154) but reformulates their criteria for semantic content analysis.

ponents of the model. For semantic content analysis, reliability primarily concerns the text model and has several aspects:

- **Stability:** Stability refers to the temporal stability of preparing text for machine parsing. When the *same* coder prepares input text at different times, the resulting text model may reflect several different coding practices within the processing model. Stability should encompass *classificational stability*, or the effects on classification of variations in coding practices. Instability can arise as coders make texts literal and explicit. Allowing multiple word senses within part of speech reduces the changes to the text, and therefore, should enhance stability.
- **Intercoder Reliability:** Although stability measures the consistency of private understandings, intercoder reliability, or reproducibility, measures the similarities and differences between different coders. Ambiguous or inconsistent coding rules for rendering text explicit can diminish reproducibility. Similarly, cognitive factors, such as situation framing or abstraction mismatches, may yield different codings. As in stability, the concern here is with the impact of coding on the possible classifications.
- **Accuracy:** Accuracy of coding is the extent to which preparing text for machine parsing conforms to the implemented processing model. For example, expecting a "literal and explicit" system to resolve metaphors would be an inaccurate coding. Similarly, exceeding grammatical coverage could lead to spurious syntactic analyses.

Semantic content analysis, as conceived here, should perform well in terms of stability and intercoder reliability precisely because it reduces the amount of coding humans must do.

Validity concerns the design of lexical classifiers and the interpretation of results. Some considerations are:

- **Construct Validity:** Construct validity refers to the correlation of different predictors for the same category instance. Two types of correlation merit attention:

  1. **Convergent constructs** use different classifiers to identify same underlying concept;
  2. **Divergent constructs** use mutually exclusive classifiers, checking to ensure that only one is present.

  In semantic content analysis, construct validity can span both the coding process and lexical classification. By retaining different surface statements, lexical interpretive semantics avoids overloading semantically canonical encodings – an inherent problem for semantic universalism. Thus, good coding practice seeks to retain alternate realizations of concepts while extending lexical classifiers to identify them.
- **Hypothesis Validity:** Hypothesis validity is the extent to which the text model yields classifications conforming to the substantive theory and the procedural theory that the analyst purports to test. For example, given a theory of organizational

decision-making and text models for a specific organization, the semantic content analysis should produce classifications consistent with both. To the extent that it does not, there may be validity problems for the text model or the classifiers or the theory of decision-making (assuming the accuracy of the text model and classifiers). Hypothesis validity raises the difficult problems of establishing correspondence between the model and the external world.

- **Predictive Validity:** Predictive validity is the extent to which the text models yield classifications consistent with the phenomenal world. One might compare the results of semantic content analysis against the classifications of humans in order to establish the validity of the coding for the text model as well as the lexical classifiers applied to it.

Semantic content analysis depends on the reliable operation of complex software systems. Since no complex computer system is ever bug-free, it is desirable to find multiple derivations for results in order to reduce the probability that conclusions follow from bugs. Where this is not possible, the analyst should verify the critical path leading to the result. A better alternative is to devise test vectors for natural language systems that identify failures for implementors to correct and determine if they cover their advertised processing model.

Beyond coding issues, different text types may have important consequences for the validity of lexical classifiers (Hurwitz & Mallery, 1987). Propaganda or insincere texts can produce spurious results for classifiers devised for sincere texts. Insincere texts make inferential demands beyond the processing model and would produce spurious classifications due to inaccurate coding. Even if unanticipated sources of erroneous classifications remain, at least, text modeling allows explicit, reproducible representation and analysis of texts.

3.4.2 Precedent Logics

The various types of precedential reasoning, such as precedents repeated over time, analogies, and metaphor are distinguished by the inductive distance between the target situation and the source situation (Mallery & Hurwitz, 1987). For an AI system to retrieve source situations for a given target, it is necessary to make a match specification which generalizes the target situation. The two main heuristics for symbolic generalization are:

- **Climb Class Hierarchies** to generalize relations and objects.
- **Drop Non-Essential Relations** to allow a match to succeed;

Category information is crucial for an AI system to use either of these two heuristics. Thus, the richer the classification of target and source situations the greater the likihood of connecting these through some inductive deformation of the target. Conceived as a search process, incremental generalizations of the target find ever more inductively distant precedents and analogies. Consequently, the quality of precedent search increases as the classification of semantic memory becomes richer and more fine-grained.

3.4.3 Modeling Differential Perception

Lexical classification opens the possibility of formally modeling differential classifications or "differential perceptions" (Jervis, 1976). Here is how it could work. First, assume no convergence in the lexical classification systems of actors and construct them independently. This will help prevent bias in favor of convergence. The lexical classification system can then be applied to the same texts, preferably with different background knowledge and in different belief systems. The resulting classifications may then be examined to locate regions of convergent and divergent classifications at the same or different levels of abstraction. In addition to direct comparisons of classificational behavior, it is also possible to investigate consequences for other facets of political cognition, such as precedent search. Precedent retrieval depends on inductive deformations of cases (section 3.4.2) and on classification for assignment of mappings. If different classification systems are in use by political actors, the divergences in classifications may lead to the retrieval of different precedents (even assuming the same historical record). This may, in turn, lead to differential problem framing.

3.4.4 Recognizing Argument Connectives

Computational argument analysis can provide a formal basis for regrounding political science (Alker, 1988a). Minsky (1987) suggests that arguments can be treated like proofs, except that the logical chain operates at a higher level of abstraction and makes larger, less justified steps. Lexical classification could recognize argument connectives if they were explicit. Unfortunately, most precomputational theories of argument seem to be AI-complete, or require arbitrary inferences to identify the connective relationships. Largely syntactic theories emphasizing explicit lexical markers – possibly Rescher's (1977) dialogical theory of argument which has shown political relevance (Alker, 1988b) – could be usefully recognized. This kind of application for lexical classification shades into parsing semantic structures, and in principle, could even extend to complete deductive inference (McAllester & Givan, 1989).

4 Conclusions

Lexical classification for text models yields the new methodology of semantic content analysis. Analysts can use this method to rigorously and reproducibly simulate classifications in political texts and political action. For international politics, the method can support studies of how convergent and divergent classification figure in conflict and cooperation. Its hermeneutic grounding in interpretive semantics anticipates differential interpretation as it insulates against distortions originating from the modeling tool itself. Since classification begins from an eidetic representation grounded in the words and grammatical relations the original text, it insulates against analyst bias; though the analyst may overlay his theoretical vocabulary on the phenomena, that vocabulary does not provide the ultimate ground of the text model. This increases the validity of analyses.

More generally, text modeling provides a new representational foundation to formal models in political science. Because this foundation is ontologically and epistemologically neutral, it can support culturally, ideologically, and politically neutral analysis. Future research may extend text modeling from recognition and generation of arguments to a new symbolically-grounded decision science. This decision science would build from a structural theory of rationality whose inferential capabilities draw from research to create unified theories of cognition (Newell, 1989) but whose noetic foundation is in text modeling. Access to the gnoosphere (collective human knowledge) through the development of encyclopedic knowledge bases (Lenat & Guha, 1990), especially text models derived from the literature on international politics, could provide grist for the mill. As these 21st century descendants of text modeling come to support evolutionary, cognitively-informed world system models, the emerging social science workbench may come to be an indispensable research associate for political scientists.

5 Acknowledgments

This paper was improved by comments from Carl Hewitt, Valerie M. Hudson, Robert P. Weber, and Jeremy M. Wertheimer. The implementation follows a path pioneered by Boris Katz and Patrick Winston. Gavan Duffy's parser helped make this research possible. The encoding of syntactic parses built from the Winston-Katz research and evolved from debates with Duffy about the correct division of labor between syntax and its description in logical form. Analysis of game protocols with Roger Hurwitz since 1985 motivated the development of the lexical classification system reported here. The moral support and foresight of Hayward R. Alker, Jr. allowed imagination to become reality. Marvin Minsky, Carl Hewitt, Berthold Horn, Gerald Sussman and countless past and present members of the M.I.T. Artificial Intelligence Laboratory made up the unique discursive background that situates this research. Lincoln Bloomfield encouraged sober applications to practical problems. Any shortcomings are the sole responsibility of the author.

The author was partially supported by a National Science Foundation Presidential Young Investigator Award number DDM-8957464 to D. Sriram, Department of Civil Engineering, M.I.T. Some earlier research was partially funded by a John D. and Catherine R. MacArthur Foundation grant for research on international security and arms control to the M.I.T Center for International Studies. This paper describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the M.I.T. Artificial Intelligence Laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the United States Department of Defense under Office of Naval Research contract number N00014-85-K-0124. RELATUS research has also been done at American Microsystems, Inc. and Symbolics, Inc.

6 References

H. R. Alker Jr., [1975], "Polimetrics: Its Descriptive Foundations," in F. Greenstein and N. Polsby, eds, *Handbook of Political Science, volume 7*, Reading: Addison-Wesley, 1975: 140-210.

H. R. Alker Jr., [1984], "Historical Argumentation and Statistical Inference: Towards More Appropriate Logics for Historical Research," *Historical Methods*, Summer 1984, 17 (3): 164-173.

H. R. Alker, Jr., [1985], "Reflective Resolution of Sequential Prisoner's Dilemmas," paper presented at *The 1985 Annual Meeting of Society for General Systems Research*, March 1985.

H. R. Alker, Jr., [1986], "Bit Flows, Rewrites, Social Talk: Toward More Adequate Informational Ontologies," *Proceedings of the Tokyo University Conference on Information and Its Functions*, 1988.

H. R. Alker, Jr., [1988a], "Politics as Political Argumentation: Towards a Regrounding of the Global Discipline," paper presented at *The XIVth World Congress of the International Political Science Association*, Washington, D.C., August, 1988.

H. R. Alker, Jr., [1988b], "The Dialectical Logic of Thucydides' Melian Dialogue," *American Political Science Review*, 82(3), 1988: 805-820.

H. R. Alker Jr., J. Bennett, and D. Mefford, [1980], "Generalized Precedent Logics for Resolving Insecurity Dilemmas," *International Interactions*, 1980, 7: 165-206.

H. R. Alker Jr., and C. Christensen, [1972], "From Causal Modeling to Artificial Intelligence: The Evolution of a UN Peace-Making Simulation," in J. La Ponce and P. Smoker, editors, *Experimentation and Simulation in Political Science*, Toronto: University of Toronto Press, 1972.

H. R. Alker, Jr., G. Duffy, R. Hurwitz, and J. C. Mallery, [1990], "Text Modeling for International Politics: A Tourist's Guide to RELATUS," V. M. Hudson, ed., *Artificial Intelligence and International Politics,* Boulder: Westview Press, 1991. This a revised version of H. R. Alker, Jr., and J. C. Mallery, "From Events Data to Computational Histories: A RELATUS-based Research Program on the Collective Management of International Conflict," paper at the *The 1988 Meeting of The International Studies Association*, April, 1988.

H. R. Alker, Jr. and R. Hurwitz, [1980], *Resolving Prisoner's Dilemmas*, Washington: American Political Science Association, 1980.

H. R. Alker, Jr., R. Hurwitz, J. C. Mallery, and F. Sherman, [1989], "SHERFACS Data and The Butterworth Narrative for the Caribbean/Cuban Missile Crisis in The Light of New Evidence: More Powerful Textual Modeling Possibilities and Changing Scholarly Purposes," paper presented at the *The 1989 Joint Annual Convention of the British International Studies Association and the International Studies Association*, London, England, March, 1989.

H. R. Alker, Jr. and A. Tanaka, [1981], "Resolutional Possibilities in 'Historical' Prisoner's Dilemmas," paper presented at the *1981 Annual Meeting of the International Studies Association*, Philadelphia, March, 1981.

R. Axelrod, [1984], *The Evolution of Cooperation*, New York: Basic Books, 1984.

E. E. Azar, [1982], *The Codebook of The Conflict and Peace Data Bank*, College Park: Center for International Development, University of Maryland, 1982.

K. Bach and R. M. Harnish, [1979], *Linguistic Communication and Speech Acts*, Cambridge: M.I.T. Press, 1979.

L. W. Barsalou and G. H. Bower, [1984], "Discrimination Nets as Psychological Models," *Cognitive Science*, 1984, 8: 1-26.

J. Barwise and J. Perry, [1983], *Situations and Attitudes*, Cambridge: M.I.T. Press, 1983.

J. P. Bennett, [1987], "Deterrence in Contemporary Policy Debates: The Expression of Strategic Doctrines in Political Arguments about SDI," paper presented at *The 1987 Meeting of the International Studies Associations*, Washington, D. C., April, 1987.

R. C. Berwick and A. S. Weinberg, [1984], *The Grammatical Basis of Linguistic Performance*, Cambridge: M.I.T. Press, 1984.

R. L. Butterworth with M. E. Scranton, [1976], *Managing Interstate Conflict, 1945-74*, New York: Knopf, 1976.

R. L. Butterworth, [1980], *Managing Interstate Conflict, 1945-79*, State College: Department of Political Science, Pennsylvania State University, final report to ARPA, February, 1980.

D. Chapman, [1985], "Planning for Conjunctive Goals," *Artificial Intelligence*, July, 1987 32: 333-377.

E. Charniak and D. McDermott, [1985], *Introduction to Artificial Intelligence*, Reading, MA: Addison-Wesley, 1985.

N. Chomsky, [1965], *Aspects of a Theory of Syntax*, Cambridge: M.I.T. Press, 1965.

F. R. Dallmayr, [1984], *Language and Politics: Why Does Language Matter to Political Philosophy?*, Notre Dame: University of Notre Dame Press, 1984.

L. G. De Michael, [1989], "Overview: The Common Lisp Object System," *Lisp and Symbolic Computation*, January, 1989, 1(3/4): 227-244.

E. A. Devereux, [1989], "Computational Analysis of Similarities and Differences in Textual Descriptions: A Comparison of Newspaper Coverage of the 1967 Election in South Vietnam," paper presented at *The 1989 annual Meeting of the American Political Science Association*, Atlanta, August, 1989.

K. W. Deutsch, [1953], *Nationalism and Social Communication: Am Inquiry into the Foundations of Nationality*, second edition, Cambridge: M.I.T. Press, 1966. The first edition was originally published in 1953.

G. Duffy, [1986], "Categorial Disambiguation," *AAAI-86: Proceedings of the American Association for Artificial Intelligence*, August 1986: 1057-1061.

G. Duffy, [1987], *Language, Politics, and Method: Grounding a Computational Hermeneutic*, Cambridge: Doctoral Dissertation, Department of Political Science, M.I.T., August, 1987.

G. Duffy, [1988], "Theory and Text: How Textual Analysis Can Help Build International Relations Theory," paper presented at *The Annual Meeting of the International Studies Association*, St. Louis, March, 1988.

G. Duffy, [1989], "The Validity Problem of Content Analysis," paper presented at *The Annual Meeting of the American Political Science Association*, Atlanta, August, 1989. This is a revised chapter from (Duffy, 1987: 21-40).

G. Duffy and J. C. Mallery, [1984], "Referential Determinism and Computational Efficiency: Posting Constraints from Deep Structure," *AAAI-84: Proceedings of the 1984 National Conference on Artificial Intelligence*, August, 1984: 101-105.

G. Duffy and J. C. Mallery, [1986], "RELATUS: An Artificial Intelligence Tool for Natural Language Modeling," paper presented at *The 1986 Meeting of The International Studies Association*, Anaheim, March, 1986.

E. A. Feigenbaum and H. A. Simon, [1984], "EPAM-like Models of Recognition and Learning," *Cognitive Science*, 1984, 8: 305-336.

J. A. Fodor, M. F. Garrett, E. C. T. Walker, and C. H. Parkes, [1980], "Against Definitions," *Cognition*, 1980, 8: 263-367.

D. Gentner and R. Landers, [1985], "Analogical Reminding: A Good Match is Hard to Find," paper presented to the panel on "Commonsense Reasoning" at *the International Conference on Systems, Man and Cybernetics*, Tucson, Arizona, November, 1985.

A. L. George, [1959], *Propaganda Analysis: A Study of Inferences Made from Nazi Propaganda in World War II*, White Plains, NY: Row, Peterson, & Co., 1959.

H. P. Grice, [1975], "Logic and Conversion," in P. Cole, and J.L. Morgan, eds, *Syntax and Semantics 3: Speech Acts*, New York: Academic Press, 1975, 41-58.

H. P. Grice, [1978], "Further Notes on Logic and Conversation," in P. Cole, ed, *Syntax and Semantics 9: Pragmatics*, New York: Academic Press, 1978, 113-28.

J. Habermas, [1979], "What is Universal Pragmatics?," in J. Habermas, *Communication and The Evolution of Society*, T. McCarthy, translator, Boston: Beacon Press, 1979.

J. Habermas, [1981], *The Theory of Communicative Action: Reason and the Rational-*

ization of Society, Thomas McCarthy, translator, Boston: Beacon Press, 1981. German edition published in 1981.

R. Harre, [1985], *Personal Being*, Cambridge: Harvard University Press, 1985.

J. A. Hobbs, [1979], "Coherence and Coreference," *Cognitive Science*, 1979, 3:67-90.

J. A. Hobbs, [1986], "On The Coherence and Structure of Discourse," in L. Polanyi, ed *The Structure of Discourse*, Norwood, N.J.: Ablex, 1986.

O. Holsti, [1968], "Content Analysis," in G. Lindzey and E. Aronson, editors, *The Handbook of Social Psychology, Volume 2*, Reading: Addison-Wesley, 1968.

R. Hurwitz, [1990], *Re: Solving the Prisoner's Dilemma*, Cambridge: Doctoral Dissertation, Department of Political Science, M.I.T., forthcoming, 1990.

R. Hurwitz and J. C. Mallery, [1987], "RELATUS Users Notes," Cambridge: M.I.T. Artificial Intelligence Laboratory, draft, 1987.

R. Hurwitz and J. C. Mallery, [1989], "Norms in Sequential Prisoner's Dilemma Protocols: Using New Techniques of Semantic Content Analysis to Identify Normative Orientations in Computer Models of Text," paper presented at *The 1989 Annual Meeting of the American Political Science Association*, Atlanta, August, 1989.

R. Hurwitz, J. C. Mallery, H. R. Alker, Jr., and G. Duffy, [1986], "Anarchy or Community: A Study of Developmental Patterns in SPD Games," paper at *The 1986 Meeting of The International Studies Association*, Anaheim, March, 1986.

R. S. Jackendoff, [1972], *Semantic Interpretation in Generative Grammar*, Cambridge: M.I.T. Press, 1972.

R. Jervis, [1976], *Perception and Misperception in International Politics*, Princeton: Princeton University Press, 1976.

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

B. Katz, [1980], "A Three Step Procedure for Language Generation," Cambridge: M.I.T. Artificial Intelligence Laboratory AI Memo No. 599, 1980.

B. Katz, [1988], "Text Processing with the START Natural Language System," in E. Barrett, ed, *Text, Context, and HyperText: Writing with and for the Computer*, Cambridge: M.I.T. Press, 1988: 55-77. An earlier version was published as "Using English for Indexing and Retrieving," *Proceedings of the Conference on User-oriented Content-based Text and Image Handling, RIAO '88*. A longer version was appears M.I.T. Artificial Intelligence Laboratory AI Memo No. 1096, October, 1988.

B. Katz and P. H. Winston, [1982], "A Two-way Natural Language Interface," in P. Degano & E. Sandewall, eds, *Integrated Interactive Computing Systems*, Amersterdam: North-Holland, 1982.

J. J. Katz and J. A. Fodor, [1963], "The Structure of a Semantic Theory," *Language*, 1963, 39 (2): 170-210.

K. Konolige, [1984], *A Deduction Model of Belief and its Logics*, Menlo Park: Artificial Intelligence Center, SRI International, Technical Note 326, 1984.

K. Krippendorf, [1980], *Content Analysis: An Introduction to its Methodology*, Beverly Hills: Sage, 1980.

H. D. Lasswell, [1927], *Propaganda Technique in World War I*, Cambridge: M.I.T. Press, 1971.

H. D. Lasswell, [1935], *World Politics and Personal Insecurity*, New York: The Free Press, 1965.

H. D. Lasswell, N. Leites, ***et al.*****, [1949]**, *Language of Politics: Studies in Quantitative Semantics*, Cambridge: M.I.T. Press, 1949.

H. D. Lasswell and J. Z. Namenwirth, [1969], *The Lasswell Value Dictionary*, New Haven: Yale University Press, 1969.

D. B. Lenat and R. V. Guha, [1990], *Building Large Knowledge Based Systems: Representation and Inference in the Cyc Project*, Reading: Addison-Wesley, 1990.

S. C. Levinson, [1983], *Pragmatics*, Cambridge: Cambridge University Press, 1983.

J. Lyons, [1977], *Semantics*, Vol. 1, Cambridge: Cambridge University Press, 1977.

D. McAllester and R. Givan, [1989], "Taxonomic Syntax for First Order Inference," Cambridge: M.I.T. Artificial Intelligence Laboratory AI Memo No. 1134, June, 1989.

A. S. Maida & S. C. Shapiro, [1982], "Intensional Concepts in Propositional Semantic Networks," *Cognitive Science*, 1982, 6: 291-330.

J. C. Mallery, [1985], "Universality and Individuality: The Interaction of Noun Phrase Determiners in Copular Clauses," *Proceedings of The Association for Computational Linguistics,* 1985.

J. C. Mallery, [1987], "Computing Strategic Language: Natural Language Models of Belief and Intention," paper presented at the *The 1987 Meeting of The International Studies Association*, April, 1987.

J. C. Mallery, [1988a], "Thinking About Foreign Policy: Finding an Appropriate Role for Artificially Intelligent Computers," Cambridge: Master's Thesis, M.I.T. Department of Political Science, 1988. A revised version was presented at the *The 1988 Meeting of The International Studies Association*, April, 1988.

J. C. Mallery, [1988b], "Beyond Correlation: Reconstructing the Relational Structure of Conflict Processes from SHERFACS Data," paper presented at the *The 1988 Meeting of The International Studies Association*, April, 1988.

J. C. Mallery, [1988c], "Identifying Implicit Categories in Text Models: A Step in Computational Hermeneutics," paper presented at the *XIVth World Congress of the International Political Science Association,* Washington, D.C., August, 1988.

J. C. Mallery, [1989], "The Invertibility Principle of Knowledge Representation," Cambridge: M.I.T. Artificial Intelligence Laboratory, draft.

J. C. Mallery, [1990], "Constraint-Interpreting Reference," Cambridge: M.I.T. Artificial Intelligence Laboratory, AI Memo No. 827, forthcoming. A draft version of this memo has seen limited circulation since March, 1985.

J. C. Mallery and G. Duffy, [1990] "A Computational Model of Semantic Perception," Cambridge: M.I.T. Artificial Intelligence Laboratory, AI Memo No. 799, forthcoming. A draft version of this memo has seen limited circulation since January, 1986.

J. C. Mallery and R. Hurwitz, [1987], "Analogy and Precedent in Strategic Decision-Making: A Computational Approach," paper presented at *The Annual Meeting of the American Political Science Association*, September, 1987.

J. C. Mallery, R. Hurwitz, and G. Duffy, [1987], "Hermeneutics," in S. C. Shapiro, *The Encyclopedia of Artificial Intelligence*, New York: John Wiley, 1987: 362-376.

J. C. Mallery, R. Hurwitz, with H. R. Alker, Jr., and G. Duffy, [1986], "Analyzing Natural Language Protocols of Sequential PD Games," paper presented to the Microcomputer Group at *The 1986 Meeting of the American Political Science Association*, Washington, D.C., August, 1986.

T. W. Malone, K. R. Grant, K. R. Lai, R. Rao, and D. A. Rosenblit, [1987], "Semi-Structured Messages are Surprisingly Useful for Computer-Supported Coordination," *ACM Transactions on Office Information Systems*, 1987, 5: 115-131.

M. P. Marcus, [1980], *A Theory of Syntactic Recognition for Natural Language*, Cambridge: M.I.T. Press, 1980.

D. Mefford, [1987], "Analogical Reasoning and The Definition of the situation: Back to Snyder for Concepts and Forward to Artificial Intelligence for Method," in C. F. Hermann, C. W. Kegely, Jr., and J. R. Rosenau, eds, *New Directions in The Study of Foreign Policy*, Boston: Allen & Unwin, 1987: 221-244.

D. Mefford, [1988a], "Women, Fire, and Artificial Intelligence," paper presented at the colloquium "Words, Deeds, and Social Things," Lincare College, Oxford University, March 11-13, 1988.

D. Mefford, [1988b], "An Alternative to Rational Choice: The function of Stories in the Formulation of Foreign Policy with Examples from U.S. Interventions," Columbus: Department of Political Science, The Ohio State University, unpublished paper, summer, 1988.

R. S. Michalski, [1987], "Concept Learning," in S. C. Shapiro, *The Encyclopedia of Artificial Intelligence*, New York: John Wiley, 1987: 185-194.

T. M. Mitchell, R. M. Keller, and S. T. Kedar-Cabelli, [1986], "Explanation-Based Generalization: A Unifying View," *Machine Learning*, 1986, 1: 47-80.

M. Minsky, [1987], *The Society of Mind*, New York: Simon and Schuster, 1987.

R. C. Moore, [1985], " A Formal Theory of Knowledge and Action," Palo Alto: Center for Study of Language and Information, Stanford University, report no. CSLI-85-31, 1985.

J. Z. Namenwirth and R. P. Weber, [1987], *Culture Indicators Research: A Content Analytic Perspective*, Boston: Unwin Hyman, 1987.

A. Newell, [1980], "Physical Symbol Systems," *Cognitive Science*, 1980, 4: 135-183.

A. Newell, [1989], *The 1987 William James Lectures: Unified Theories of Cognition*, Cambridge: Harvard University Press, forthcoming.

A. Ortony, G. L. Clore, and M. A. Foss, 1987], "The Referential Structure of the Affective Lexicon," *Cognitive Science*, July-September, 1987, 11(3): 341-364.

J. Piaget, [1979], *The Child's Conception of the World*, J. Tomlinson and A. Tomlinson, trans., Totowa: Littlefield Adams, 1979.

C. S. Peirce, [1901], "Abduction and Induction," reprinted in J. Buchler, *Philosophical Writings of Peirce*, New York: Dover Publications, 1955. Written circa. 1901.

I. S. Pool, H. D. Lasswell, D. Lerner, *et al.*, **[1970]**, *The Prestige Press: A Comparative Study of Political Symbols*, Cambridge: M.I.T. Press, 1970.

N. Rescher, [1977], *Dialectics: A Controversy-Oriented Approach to the Theory of Knowledge*, Albany: State University of New York Press, 1977.

N. Rescher, [1978], *Peirce's Philosophy of Science: Critical Studies in His Theory of Induction and Scientific Method*, Notre Dame: University of Notre Dame Press, 1978.

P. Ricoeur, [1971], "The Model of Text: Meaningful Action Considered as Text," *Social Research*, 1971, 38: 529-62. Reprinted in J. B. Thompson, editor and translator, *Paul Ricoeur: Hermeneutics and The Human Sciences*, Cambridge: Cambridge University Press, 1981.

P. Ricoeur, [1975], "Phenomenology and Hermeneutics," Translated and reprinted in J. B. Thompson, ed and trans, *Paul Ricoeur: Hermeneutics and the Human Sciences*, Cambridge University Press, Cambridge, 1981. Originally published as "Phenomenologie et Hermeneutique," *Phanomenologische Forschungen*, vol. 1, E. W. Orth, ed, Karl Alber, Freiberg, 1975, 31-77.

P. Ricoeur, [1977], "Metaphor and the Semantics of Discourse," in *The Rule of Metaphor: Multi-Disciplinary Studies in Metaphor*, translated by R. Czerny *et al*, Toronto: University of Toronto Press, 1977, 65-100.

B. Russell, [1948], *Human Knowledge: Its Scope and Limits*, New York, Simon and

Schuster, 1948.

R. C. Schank, [1972], "Conceptual Dependency: A Theory Of Natural Language," *Cognitive Psychology*, 3, 552-631 (1972).

R. C. Schank and R. P. Abelson, [1977], *Scripts, Plans, Goals, and Understanding*, Hillsdale, N.J.: Lawrence Erlbaum, 1977.

J. R. Searle, [1969], *Speech Acts*, Cambridge: Cambridge University Press, 1969.

J. R. Searle, [1979], "Metaphor," in Ortony, Andrew, ed, *Metaphor and Thought*, Cambridge: Cambridge University Press, 1979: 92-123.

M. J. Shapiro, [1984], *Language and Politics*, New York: New York University Press, 1984.

M. J. Shapiro, [1988], *The Politics of Representation*, Madison: University of Wisconsin Press, 1988.

F. L. Sherman, [1987a], *Partway to Peace: The United Nations and The Road to Nowhere*, State College: Doctoral Dissertation, Department of Political Science, Pennsylvania State University, 1987.

F. L. Sherman, [1987b], "Four Major Traditions of Historical Events Research: A Brief Comparison," paper presented to the M.I.T./D.D.I.R Conference on "New Directions for Storing, Indexing, Retrieving, Coding and Analyzing Information on International Events," Cambridge: Center for International Studies, M.I.T., November, 1987.

F. L. Sherman, [1988], "SHERFACS: A New Cross-Paradigm, International Conflict Dataset," paper presented at *The 1988 Meeting of The International Studies Association*, St Louis, Missouri, March, 1988.

H. A. Simon, [1962], "The Architecture of Complexity," *Proceedings of The American Philosophical Society*, December, 1962, 106:467-482. Reprinted in H. A. Simon, *The Sciences of The Artificial*, second edition, Cambridge: M.I.T. Press, 1981. Originally published in 1969.

H. A. Simon, [1985], "Human Nature in Politics: The Dialogue of Psychology with Political Science," *The American Political Science Review*, 1985, 79: 293-304.

P. J. Stone, D. C. Dunphy, M. S. Smith, D. M. Ogilvie, with associates, [1966], *The General Inquirer: A Computer Approach to Content Analysis*, Cambridge: M.I.T. Press, 1966.

Symbolics, Inc., [1988], *Reference Manual for Symbolics Computers*, Cambridge: Symbolics, Inc., 1986. Revised 1987, 1988.

C. Taylor, [1985], *Human Agency and Language: Philosophical Papers*, Cambridge: Cambridge University Press, 1985.

D. L. Waltz, [1975], *Understanding Line Drawings of Scenes with Shadows*, in P. H.

Winston, ed, *The Psychology of Computer Vision*, New York: McGraw-Hill, 1975: 19-91.

R. P. Weber, [1985], *Basic Content Analysis*, Beverly Hills: Sage, 1985.

T. Winograd, [1971], *Procedures as a Representation for Data in a Computer Program for Understanding Natural Language*, Cambridge: Doctoral thesis, Department of Electrical Engineering and Computer Science, M.I.T., 1971.

T. Winograd and F. Flores, [1986], *Understanding Computers and Cognition: A New Foundation for Design*, Norwood, NJ: Ablex, 1986.

P. H. Winston, [1980], "Learning and Reasoning by Analogy," *Communications of the ACM*, December, 1980, 23(12).

P. H. Winston, [1984], *Artificial Intelligence*, Reading: Addison-Wesley, 1984.

L. Wittgenstein, [1953], *Philosophical Investigations*, New York: MacMillan, 1958. First edition published in 1953.