A Lexicon for Knowledge-Based MT

Boyan Onyshkevych
US Department of Defense
and Carnegie Mellon University
Sergei Nirenburg
New Mexico State University

Abstract. In knowledge-based machine translation (KBMT), the lexicon can be specified and acquired only in close connection with the specification and acquisition of the world model (ontology) and the specification of the text meaning representation (interlingua) language. The former supplies the atoms for the specification of text meaning and provides world knowledge to support the inference processes necessary for a variety of disambiguation and meaning assignment operations. The latter is necessary for the formulation of the semantic zone of the lexicon entries, which can be viewed as containing the static building blocks of the text meaning representation. This is the view taken in the Mikrokosmos KBMT project.

1. Introduction.

Over the past decade, the number and diversity of experiments in Knowledge-Based Machine Translation (KBMT) has grown significantly (*cf.* Nirenburg *et al.*, 1987; Cullingford and Onyshkevych, 1987; Carbonell *et al.*, 1992, or Nyberg and Mitamura, 1992; Nirenburg *et al.*, 1992; etc.) It is no longer appropriate simply to state that a system adheres to the KBMT paradigm. Further explanations are necessary.

The KBMT approach in the Mikrokosmos project[1] can be briefly summarized as follows (for a more detailed description of this approach see Nirenburg *et al.*, 1992). In the most general terms, the method to which we adhere is similar to other KBMT approaches: given a source language text, extract and represent its meaning in a language-neutral format (the interlingua), thus transforming the input text into an interlingua text; next apply a target language generator to the interlingua text to produce a target language text.

The differences among the various KBMT approaches mostly have to do with the intended coverage of the interlingua and the depth to which it analyzes the source text. Let us consider just two examples. In some practical applications (e.g., the KANT system, Carbonell *et al.*, 1992, or Nyberg and Mitamura, 1992), the interlingua is built for an MT system which involves pre-editing of the source text, so that only a limited vocabulary and a subset of the source language syntax are used. In such a limited-coverage situation, there is an opportunity to avoid reference to a detailed ontology and to rely on a large, though uninterpreted, set of semantic primitives.

The second such example, namely the work by Dorr (1993) and associates, is an experiment in meaning analysis which stops short of full reliance on world knowledge. It is a computational ap-

[1] The MIKROKOSMOS project is a joint research project involving the DoD and NMSU facilities of the authors; the goal of the project is to the semantic representation, knowledge, and reasoning to support the next-generation semantics-based KBMT systems.

plication of Jackendoff's theory of lexical-conceptual structure (LCS). The goal of the study is to determine whether LCSs are sufficient to serve as a pivot for MT. LCSs are structures in which the metalanguage of semantic description is not independently motivated. They are organized along syntactic dependency lines, featuring a small number of case roles with limited selectional restriction information listed using elements of the natural language in question as interlingua elements (for criticism of Jackendoff's approach, see Wilks, 1992). In its semantic part, this approach unwittingly follows the early AI NLP approaches (e.g., Schank, 1973 or Wilks, 1973), thus repeating what proved to be the weak points of those approaches, for instance, the insistence on using a small and closed set of meaning primitives. If LCSs must be used as interlinguas for MT, they need to be seriously modified, in fact, made much less LCS-like. Dorr and her associates have, in fact, undertaken this task and are at this moment getting closer to acknowledging the role of world knowledge and language-independent representation schemata (Dorr *et al.*, 1994, Dorr and Voss, 1994).

As will become clear from the discussions below, the lexicon is the pivotal static knowledge source in the KBMT approach. It mediates between the representation of the meaning of a text and the ontology. It also helps to integrate syntactic and semantic information about the text. This paper will be devoted to the explanation of why the intimate interconnection of the meaning representation, ontology, and the lexicon is essential for KBMT and how this connection is accomplished in Mikrokosmos.

The main issue discussed in this paper is *how to represent the meaning of word senses*. Of all the KBMT knowledge sources (the *static* knowledge sources -- the meaning representation language, ontology, and the lexicon, and the dynamic knowledge sources -- the results of syntactic analysis and semantic analysis), we only address the lexicon in any significant depth here, although we touch upon the others in order to illustrate the interaction of the lexicon with the rest of the modules of the paradigm.[1]

2. Background

We view the basic process of text meaning extraction as follows. First, we define the format in which the results of the analysis process is represented (the interlingua), known in the Mikrokosmos project as the *Text Meaning Representation* (TMR) language; the results of analysis of concrete texts are represented in this language and are called simply TMRs. Our TMR language is frame-based, with frames denoting, in first approximation, instances of ontological concepts, and frame slots denoting properties of these concepts.

Source language lexical units come in three varieties: those causing the appearance of new frames in the TMR (e.g., most verbs and many nouns); those finding fillers for certain slots in such frames (e.g., many adjectives and some nouns); and others, whose traces in TMR are more indirect (see examples below). Indeed, some of the TMR slot fillers are determined using any combination of semantic, syntactic, stylistic, pragmatic, and other knowledge about the source text. All lexical units are heads of entries in the MT lexicon for the source language. The semantic zone of these entries typically contains references to individual concepts in the ontology, and can be viewed as containing the static building blocks of the TMR (which is a dynamic construct).

[1] The work reported in this paper is an enhancement on the KBMT approach described, e.g., in Nirenburg et al., 1992. The lexical-semantic descriptions in our approach have gradually grown more detailed, as a variety of microtheories have been added.

As illustrated in Figure 2A, the knowledge necessary for building the component element of

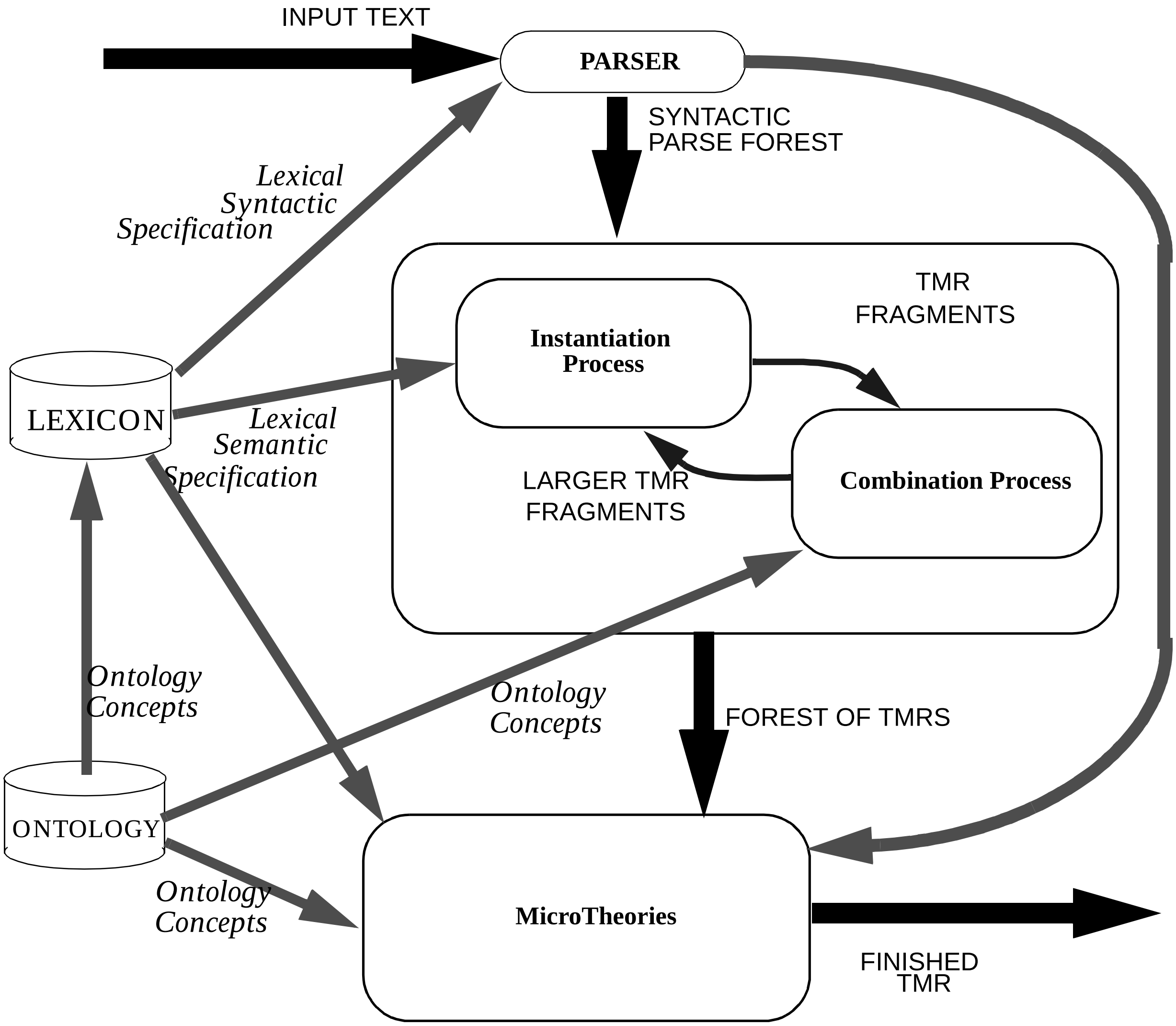


Figure 2A. Overall data flow of the architecture

specific TMRs (represented by the Instantiation process) is, in most cases, obtainable from two sources: a) the syntactic parse of the input text, and b) those parts of the lexicon which deal with syntax-semantics interface (mapping or linking) and meaning. Given all this information, initial TMR entities are determined, and relationships among them established. Further processes (this time, the background knowledge typically includes, in addition to the above sources, also the nascent TMR itself, as well as the ontology, accessible through the same lexicon) determine reference information, expand ellipsis, establish temporal, discourse, and other relations among text elements and treat “emergencies,” such as unexpected input or inability to choose among two or more remaining candidate TMRs for a given text. For a more detailed description of the analysis process see Section 10 below.

3. Organization of the Lexicon

The lexicon for a given language is a collection of *superentries* which are indexed by the citation form of the word (represented in the **ORTH-FORM** field in our lexicon formalism). Within a superentry, individual lexemes are represented in a frame-based language (FRAMEKIT (Nyberg, 1988) in the LISP version, FRAMEPAC (Brown, 1994) in the C++ version). A superentry includes all the lexemes which have the same dictionary form, regardless of syntactic category, pronunciation, or sense. Thus, a given superentry might include any number of noun, verb, adjective, etc. lexemes.

In the examples below, lexemes ("entries") inside a superentry have names which are formatted using the character "+", followed by the citation form, followed by "-" and an indication of the syntactic category (e.g., **v**, **n**, **adj**) of the entry and its sense number. For example, **+eat-v2** introduces the entry for the second verbal sense of *eat*.

Proper names which reference specific entities in the world may also have lexical entries; in the lexical entry they reference an entry in an onomasticon (an inventory of specific named entities, see Section 6.2.2) in their lexical-semantic description (see below), but otherwise are similar to other lexicon entries. For example, **+Paris-n1** might be the label for the English lexical item *Paris* which names the city Paris, France. This arrangement allows language-independent world knowledge to be maintained independently of language-specific nomenclature (which, in turn, affects its phonology, morphology, syntactic behavior, etc.)

Since the approach to the lexicon discussed here is for the support of building a language-independent meaning representation of texts (i.e., the interlingua), the main focus of the lexicon is delivering the specific meaning representation of each lexeme. As a gross generalization, it could be said that all the other information in the lexicon merely supports the delivery and selection of the appropriate lexical entry and its meaning. The meaning of a lexical entry (represented in the **SEM-STRUC** zone; see below) is represented by a lexical semantic representation, which is constructed using atoms from the ontology (as well as some other primitives). Instantiation of the lexical semantic representations of the words in a text forms the core of the TMR.

Each lexicon entry is comprised of a number of *zones* corresponding to the various levels of lexical information. The zones containing information for use by analysis or generation components of an MT system are: **CAT** (syntactic category), **ORTH** (orthography — abbreviations and variants), **PHON** (phonology), **MORPH** (morphological irregular forms, class or paradigm, and stem variants or "principle parts"), **APPL** (dialect or other sublanguage indicators), **SYN** (syntactic features such as `attributive`), **SYN-STRUC** (indication of sentence- or phrase-level syntactic dependency, centrally including subcategorization), **SEM-STRUC** (lexical semantics, meaning representation), **LEXICAL-RELATIONS** (collocations, etc.), **LEXICAL-RULES** (listing of true positive and false positive lexical rules that appear to apply to the lexeme), and **PRAGM** (information related to pragmatics as well as stylistic factors). A special **ANNOTATIONS** zone contains ancillary user, lexicographer, and administrative information, such as modification audit trail, example sentences, printed dictionary definitions, cross-references (what other lexemes is this one referenced by), etc. Below is a fuller specification of the structure of the lexicon entry, starting with the superentry (such as **bark** which is then broken down into categories and senses, such as **+bark-v1**) and further specifying the zones and fields in each zone, in a BNF-like notation (bold text is used to identify the short forms of the zone/field names as used in the discussion):

```
<superentry> :=
          ORTHOGRAPHIC-FORM:  “form”
          ({syn-cat}: <lexeme> * ) *
<lexeme> :=
          CATEGORY: {syn-cat}
          ORTHOGRAPHY:
                    VARIANTS: “variants”*
                    ABBREVIATIONS: “abbs”*
          PHONOLOGY: “phonology”*
          MORPHOLOGY:
                    IRREGULAR-FORMS: (“form”
                                      {irreg-form-name})*
                    PARADIGM: {paradigm-name}
                    STEM-VARIANTS: (“form” {variant-name})*
          ANNOTATIONS:
                    DEFINITION: “definition in NL” *
                    EXAMPLES: “example”*
                    COMMENTS: “lexicographer comment”*
                    TIME-STAMP: date-of-entry lexicog-id
                    DATE-LAST-MODIFIED: date lexicog-id
                    CROSS-REFERENCES: lexeme *
          APPLICABILITY:
                    LOCALITY: “locale”*
                    FIELD: “field”*
                    LANGUAGE: “language”*
                    CURRENCY: “era”*
          SYNTACTIC-FEATURES:
                    SYNTACTIC-CLASS: class
                    IDIOSYNCRATIC-FEATURES: (feature value)*
          SYNTACTIC-STRUCTURE:
                    SYNTACTIC-STRUCTURE-CLASS: class
                    SYNTACTIC-STRUCTURE-LOCAL: fs-pattern
          SEMANTIC-STRUCTURE:
                    LEXICAL-MAPPING: lex-sem-specification
                    MEANING-PROCEDURE: meaning-specialist
          LEXICAL-RELATIONS:
                    PARADIGMATIC-RELS: ({p-r-type} lexeme)*
                    SYNTAGMATIC-RELS: ({s-r-type}
                                        f-struct | lexeme)*
          LEXICAL-RULES:
                    LR-CLASS: class *
                    LR-LOCAL: (LR# (lexeme | OK | NO)) *
```

```
PRAGMATICS:
          STYLISTICS: ({FORMALITY, SIMPLICITY,
                    COLOR, FORCE, DIRECTNESS,
                    RESPECT}  value) *
          ANALYSIS-TRIGGERS: trigger *
          GENERATION-TRIGGERS: trigger *
```

The **CAT**, **ORTH**, **MORPH**, **SYN**, and **SYN-STRUC** zones are used primarily during syntactic parsing stage (which in our paradigm also includes segmentation, tokenization, and morphological analysis), which is not addressed at great length in this discussion. This stage precedes semantic analysis in the simplest implementation, but may be interleaved with semantic processing in future experiments. The **SYN-STRUC** zone, discussed in further detail in Section 7, specifies local syntactic context for the lexeme for use in syntactic parsing, but also plays a crucial role in establishing bindings in the syntax-semantics interface. The **APPL** zone provides information in analysis that may be used in preferring one word sense over another (depending on the expected or identified sublanguage), and in generation in selecting from among synonyms.

The **LEX-REL** zone, currently still in preliminary development, is intended to provide reference to primarily collocational information; each collocation is categorized, e.g., using Mel’cuk-style categories (Mel’cuk, 1984), and represented in a partially-specified f-structure (of the same style as the **SYN-STRUC** specification). Since collocations are compositional in meaning (have transparent “decoding” despite the idiosyncratic “encoding”), particularly when the word senses are identified, there is typically no need to represent the semantics or pragmatics of the collocations further; if there is need, the relation is instead represented by direct reference to another lexeme. The **PAR-RELS** slot of the **LEX-REL** zone is used to represent such relations as synonymy, antonymy, or hyponymy, but primarily only as an indexing convenience; these relations are primarily reflected by the relative ontological positions of the concepts used to define each lexeme.

As our primary current research interests center on issues in semantics (in particular, lexical semantics) the **SEM-STRUC** zone attracts central attention. Through this zone the lexicon connects with the ontology and the onomasticon, thus becoming the locus of the atomic links between lexical units in texts and the language-neutral text meaning representation, or TMR. The formalism for the lexical semantic specification in the **SEM-STRUC** zone (specifically, the **LEX-MAP** field) is discussed in detail in Section 8, while the utilization of that specification is discussed in subsequent sections.

The meaning of a small number of lexemes is not representable by the instantiation of a **LEX-MAP** template. Words that fall into this category include deictics, intensifiers such as *very*, some referring expressions, and discourse markers; the effect of these words on the meaning of the utterance is not the instantiation of a predetermined concept, nor the linking of predetermined structures from the TMR. To handle the various other effects that these relatively small classes of words have on the meaning representation, alternative mechanisms are provided: the **MEAN-PROC** in the **SEM-STRUC** zone, and the **TRIGGER** slots in the **PRAGM** zone. These mechanisms allow for the invocation of functions or procedures that modulate or modify the meaning representation of an utterance, in a manner somewhat akin to Word Expert Parser specialists in (Small and Rieger, 1982). The **MEAN-PROC** allows for functions that modulate the meaning representation, as in the case of the adverbial *very*, where the function intensifies the value of a scalar attribute towards one or the other extreme. The trigger mechanisms in the **PRAGM** zone allow for the invocation of procedures or microtheories that have a particular mission; the **A-TRIG** for *the*, for example, invokes a definite reference resolution mechanism during semantic analysis (similarly deictics invoke a mechanism which attempts to identify the appropriate referent.)

4. Economy in Lexicon Specification and Acquisition

Among the central issues in current computational lexicography are “packing” information in the lexicons and facilitating acquisition. Two (connected) approaches have been followed. First, attempts have been made to cross-index information in the entries, for instance, by building lexicons as hierarchical structures, with a variety of features inherited from parents to children. Second, certain word senses might not be overtly listed in the lexicon as separate entries, instead of which instructions are supplied of how to create such entries when an application program requires them. For various pragmatic reasons, some of these senses are generated not on an as-needed basis but at acquisition time (thus nullifying the space savings in favor of improved lexicon quality).

4.1 Cross-Indexing and Inheritance

A variety of cross-indexing mechanisms are used to minimize redundancy within the lexicon. Inheritance is one type of cross-indexing used, for example, to indicate that a particular verb is of syntactic **SYN-CLASS** *basic-bitransitive*, thus avoiding the need for a syntactic specification or syntactic features to be specified locally in the corresponding entry: the information will be inherited from the specification in the definition of the class. Inheritance is used explicitly in our lexicon in the MORPH (paradigm), **SYN**, **SYN-STRUC**, and **LEX-RULES** zones. “Horizontal” cross-reference can be used to indicate that, say, the third and fourth verb sense of *eat* share the same **SYN-STRUC**, zone or that all verbal senses of *eat* share the same **PHON** and **MORPH**; this is accomplished by simple reference pointers in the underlying data structures.

4.2 Lexical Rules

In the interests of efficiency in knowledge acquisition and to capture generalizations about productive lexical alternations and derivations, lexical rules (LRs) are used to generate lexical entries dynamically from lexical entries encoded statically in the lexicon. In our model LRs can be used to cover a broad spectrum of phenomena, including syntactic alternations such as passivization and dative, regular non-metonymic and non-metaphoric meaning alternations (such as those described in Apresjan (1974) and Pustejovsky (1991)), as well as some productive derivational processes such as formation of deverbal nominals or deverbal adjectives. Thus the LRs in this paradigm include the phenomena covered by LRs in LFG in Bresnan (1982), but also many of the Lexical Inference Rules (LIRs) from Ostler and Atkins (1992), which necessarily include semantic shifts. When LRs are added to the lexicographer’s arsenal, the lexicon as a whole becomes a list of (super)entries plus a list of LRs. The discussion below highlights some *a priori* restrictions on the scope and content of LRs.

LRs in our model apply to only one lexical entry at a time and, thus, do not cover phenomena which involve two or more senses (such as compounding in German and other languages). Also not covered (by the mechanism of LRs) is the treatment of metonymy, seen procedurally as the situation when there is a violation of selectional restrictions for an entire set of senses of two or more lexical units that have to be combined in a single semantic dependency structure (which might be the case, for example, in treating *The White House announced* ...) We delegate the treatment of metonymy and similar phenomena to the processing component of the application system: a dynamic knowledge source such as a semantic interpreter or a lexical selector in generation; however, the knowledge required for these processes is in fact encoded into the ontology's network of relations or links, as well as in weights for those links (see Section 10 for a description of ontological graph search – our central mechanism for carrying out semantic analysis).[1] A useful rule of thumb for deciding whether a phenomenon should be treated in a static knowledge source (e.g., through LRs) or in a dynamic knowledge source (a processor module) is whether the phenomenon is language-specific (go with LRs) or language-neutral (treat it using processing-related rules). Thus the scope of our LRs differs slightly from Ostler and Atkins' LIRs, where they capture both language-dependent and language-independent derivations in the LIRs, while we focus our LRs on language-dependent derivations. They do, however, exclude alternations strictly based on pragmatics or world knowledge, as we do.

LRs consist of a left-hand side (LHS) which constrains the lexical entries to which the rule can apply and a right-hand side (RHS) which stipulates how the new lexical entry will differ from the original. Lexical entries which are produced by a LR are themselves eligible to match the LHS of an LR. Both sides of the LR can reference any zone of the lexical entry; typically the RHS modifies the local syntactic information and the lexical semantic specification (or at least the syntax-semantic interface). Often, however, the syntactic category, syntactic features, and orthography are affected as well (in derivational cases).

All of the lexicon zones are available to the LRs, both to the LHS for constraining the application of the rules, as well as to the RHS for modification as part of the alternation or derivation that the LR reflects. The syntactic category (i.e., the **CAT** zone) is often modified in derivational rules, e.g., in those LRs which produce nominal or adjectival forms from verbs. The syntactic features (**SYN**) and syntactic structure (**SYN-STRUC**) of a lexeme would be affected in most LRs. Particularly in derivational LRs, the word form itself changes, thus the **ORTH**ography, **PHON**ology, and **MORPH**ology of the lexeme would change. The lexical semantic representation (**LEX-MAP**) can be used to constrain the application of rules in the LHS: in the passive rule (see above), an AGENT is required in the lexical semantic specification on the LHS. Regular alternations such as those described by Beth Levin and others, for example in Levin (1989, 1991), would be constrained to apply to only those lexemes with a particular concept (or descendant of that concept) as semantic head (e.g., LOAD) in the **LEX-MAP**. Some LRs cause the semantic representation itself to change. In other cases, however, there is no actual semantic reflection of derivational LRs, because, for example, deverbal nouns and adjectives (such as *abuse* and *abusive*) are typically represented in the identical fashion as the base forms in this paradigm (perhaps only with a different syntax-semantics interface). Ostler and Atkins require changes in the semantics in the RHS of their LIRs, but the nature of the semantic representation used in this paradigm (tending to diverge substantially in predicate/argument structure from that of the surface syntax) results in no semantic change for some derivations.

The simplest mechanism of rule triggering is to include in each lexicon entry an explicit list of applicable rules. LR application can be chained, so that the rule chains must be expanded, either statically, in the specification, or dynamically, at application time. This approach avoids any inap-

[1] Some simple metonymies might be handled by LRs (with equivalent results) in a more economical manner than the semantic analysis processing provides, and thus may be "cached" by encoding them into LRs; this will be addressed by further experimentation.

propriate application of the rules (overgeneration), though at the expense of tedious work at lexicon acquisition time. The other approach is to maintain a bank of LRs, and rely on the left-hand sides to constrain the application of the rules to only the appropriate cases; in practice, however, it is difficult to set up the constraints in such a way as to avoid over- and undergeneration. For example, it is difficult to constrain the LHS to select exactly the set of verbs to which *-able* derivation applies. As another example, to prevent the passivization of idioms such as *kick the bucket* (but allow it on *spill the beans*) it is necessary to set up constraints to block application in inappropriate cases; in this case, requiring both an `AGENT` and a `THEME` (or `BENEFICIARY`, etc.) in the lexical semantics appropriately constrains the passive rule and prevents overgeneration. Related mechanisms for restricting the application of LRs to avoid overgeneration, such as blocking and preemption, have reduced effectiveness in practical situations where the lexicon is incomplete or is under construction (because the form that is supposed to block or pre-empt may not have been entered yet).

The reliance on rule application at run-time (vs. listing in the lexicon) does not allow explicit ordering of word senses, a practice preferred by many lexicographers to indicate relative frequency or salience; this sort of information can be captured by other mechanisms (e.g., using frequency-of-occurrence statistics). This approach does, however, capture the paradigmatic generalization that is represented by the rule, and simplifies lexical acquisition.

The approach adopted in Mikrokosmos (although still under development) is a hybrid approach, as a compromise of linguistic generality, processing considerations, and acquisition considerations (the full space of related issues will be addressed in a future paper). The LRs are written with the LHS attempting to constrain the forms to which the rule applies as tightly as possible (to include pre-emption in addition to constraints on **`SYN-STRUC`**s, **`SEM-STRUC`**s, etc.) At lexicon acquisition time, all applicable LRs are applied to the base form, producing full lexical entries for the derived forms. For any rules that successfully apply, the acquisition tool checks for the existence of the resulting orthographic form in a corpus; this only helps in the cases where a new dictionary form is created, and does not apply in the cases of meaning-only or subcategorization shifts. Any new senses with orthographic forms that do not appear in the corpus are summarily rejected; all the remaining senses are presented to the lexicographer for verification and/or augmentation. Additionally, the occurrences in the corpus are retained by the lexicographer as examples (in the **`ANNOTATIONS`** field). The lexicographer needs to review the corpus examples and determine the distribution of senses for that form, as is usual in lexicography. The base form maintains an inventory of LRs which apply (in the **`LR-LOCAL`** facet), along with either an indication that the result was rejected by the lexicographer (‘`NO`’), or the lexeme name of the resulting form; for some LRs which are productive and fairly regular (such as passivization), instead of storing all the derived lexical entries, the LR fields in the base forms merely indicate ‘`OK`’, and the lexical entries are produced at run time on an as-needed basis. In syntactic parsing of texts (normal processing), for any word form in the text which is not found in the lexicon, all the LRs can be attempted to try to generate the novel form, as a recovery procedure; some rules (such as passivization) are explicitly invoked by the syntactic parsing processes. As the lexicon grows, more of the highly productive LRs will be restrained from application at acquisition time, and only applied at run time, for purposes of storage economy.

5. Some Initial Examples

Before launching into a more detailed discussion of the lexicon and its support knowledge sources and their formalisms (i.e., the ontology, the syntactic f-structure, and the TMR), a few simple illustrations are in order; these examples will illustrate the particulars of how the lexical semantic specification for particular lexemes interact to form the TMR for the utterance. As our research concentrates on semantics, we do not emphasize the syntactic information in the examples. Suffice it to say that we assume a syntactic parse to a tree structure with heads projecting constituents (this will be significant in the syntax-semantics interface). Thus the examples below will focus exclusively on the **SEM-STRUC** zone, particulary the **LEX-MAP** information. The information from this zone is specified using atoms from the ontology, and will be manipulated to form the TMR for the text.

The lexemes from the example sentence *The chihuahua ate the apple* are presented in abbreviated form below (this example ignores tense, aspect, determiners, etc.) Figure 5A presents a graphical view of the lexical-semantic representation (i.e., **LEX-MAP** from the **SEM-STRUC** zone of the lexicon) for the nouns and the verb. The simplest is for *apple*: the semantics zone of this lexicon entry (for simplicity, we ignore polysemy for the time being and refer to the basic “fruit” sense of *apple*) simply indicates that there is a concept in the ontology equivalent to the meaning of this lexeme. The **%** marks a string which is an instantiation of an ontological concept.

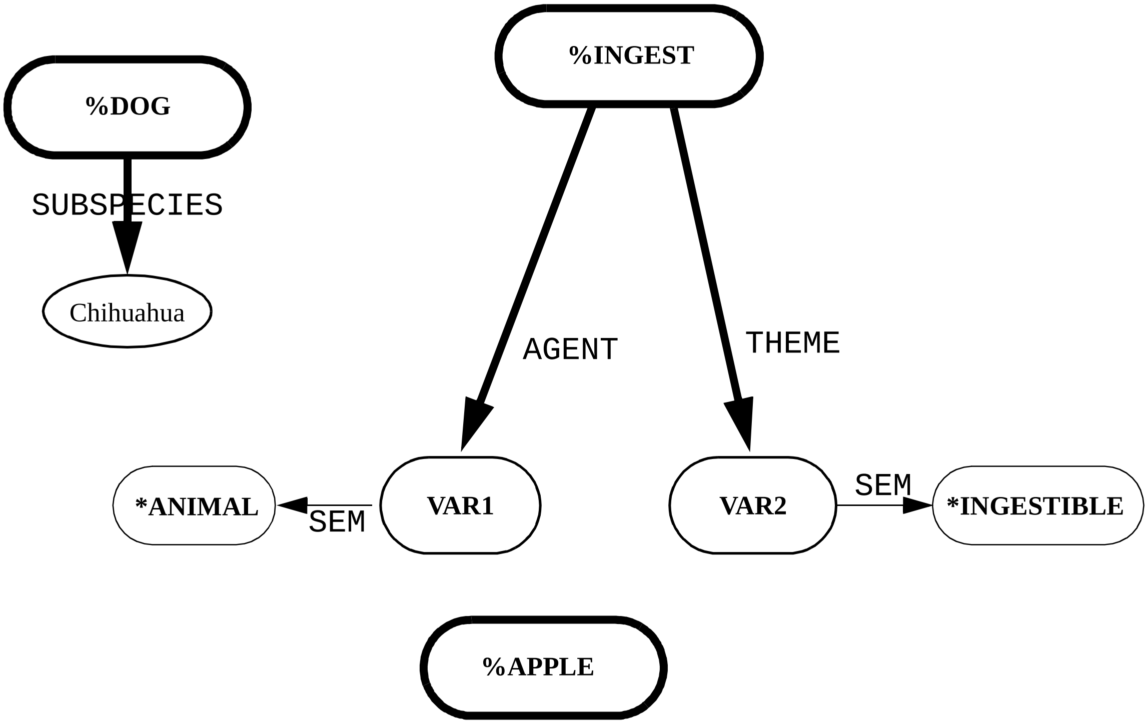


Figure 5A. Graphical representation of lexical semantics for the three words *chihuahua*, *eat*, and *apple*. The heavy vertical links represent slots (case roles) on a concept, while lighter horizontal links represent constraints on the expected/possible fillers of those slots. Note that there are three distinct structures, one for each of the three words. The * marks a reference to a concept from the ontology; the % identifies concepts which will be instantiated during the analysis process.

The representation of the other noun is somewhat different. It so happens that the ontology used to support our sample dictionary does not have a concept for chihuahuas (the question of the grain-size trade-offs in designing ontologies and lexicons is far from settled; for further discussion see Section 9). Thus, our lexicon entry for *chihuahua* contains in its **SEM-STRUC** zone a request to instantiate a DOG concept, but with further (lexicon-stipulated) specification that the dog is of the subspecies called Chihuahua.

The representation for *eat* has different complexity, as it is an argument-taking lexical unit whose semantic description must include information about building a semantic dependency structure comprising the meaning of the unit itself and the meanings of its arguments. This structure-building operation, with a concomitant disambiguation process, is supported by listing semantic constraints on the meanings of arguments of the argument-taking lexical unit. In this case, the INGEST concept has (at least) two slots, named AGENT and THEME. The semantic constraints on those slots are represented in "facets" of those slots, specifically SEM facets, represented by the lighter horizontal arrows in the diagrams above and below (the VALUE facet stores the filler of the slot, while the SEM facet stores constraints on allowed fillers of the slot). The semantic constraints are themselves concepts from the ontology; any concept (or instantiation of a concept) which falls below the constraint in the ontology tree satisfies the constraint.

During semantic analysis, all the lexical semantic specifications are instantiated, as illustrated

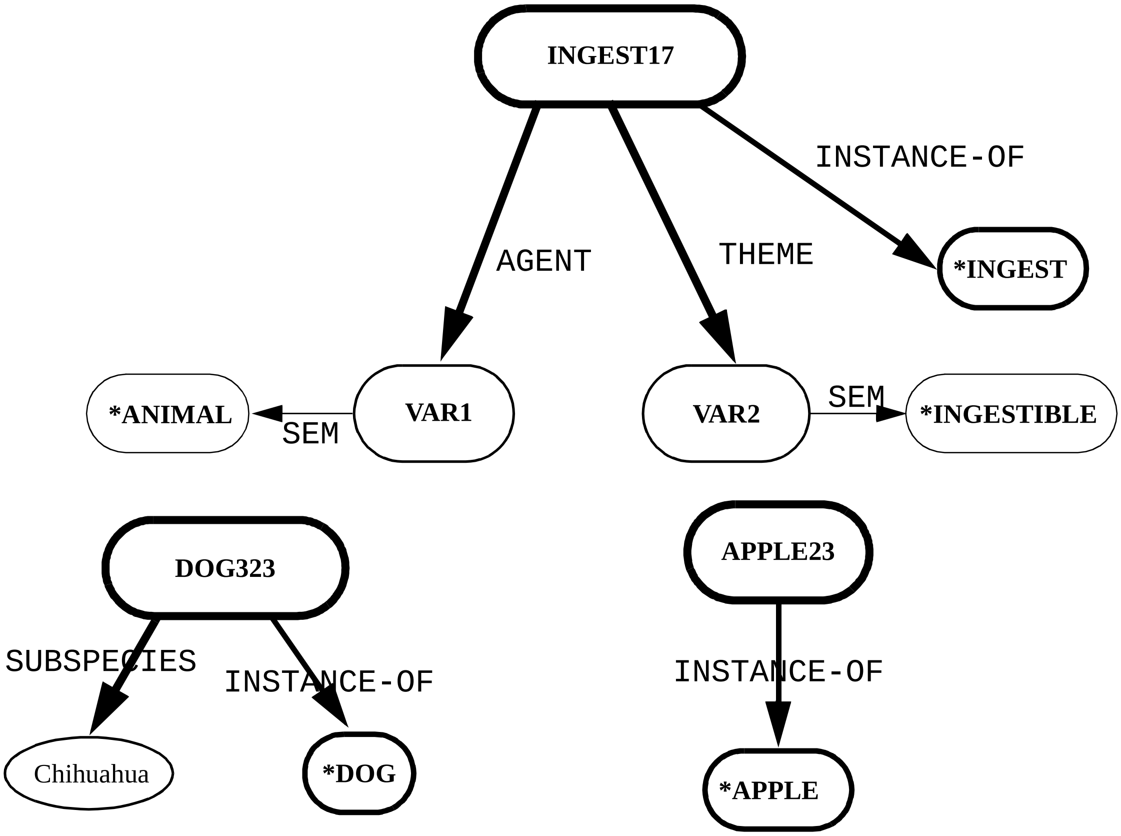


Figure 5B. Graphical representation of instantiated lexemes (3 discrete structures)

in Figure 5B, which illustrates (in a graphical form), a set of fragments of a TMR. A uniquely numbered instance of each relevant concept is created. Each instantiated concept has in its frame representation the slot INSTANCE-OF whose filler indicates from what concept this instance was produced. Note that the semantic constraints on preferred fillers of various slots, as reflected by the SEM links, remain pointers to concepts from the ontology (marked by a * in this notation), since they are not instantiated.

The instantiations are combined in well-defined ways to produce the initial TMR for the text, as illustrated diagrammatically in Figure 5C. In the system-internal representation, each instantiation remains an independent structure, with pointers (in the form of the structure name) as the filler of the relevant slot:

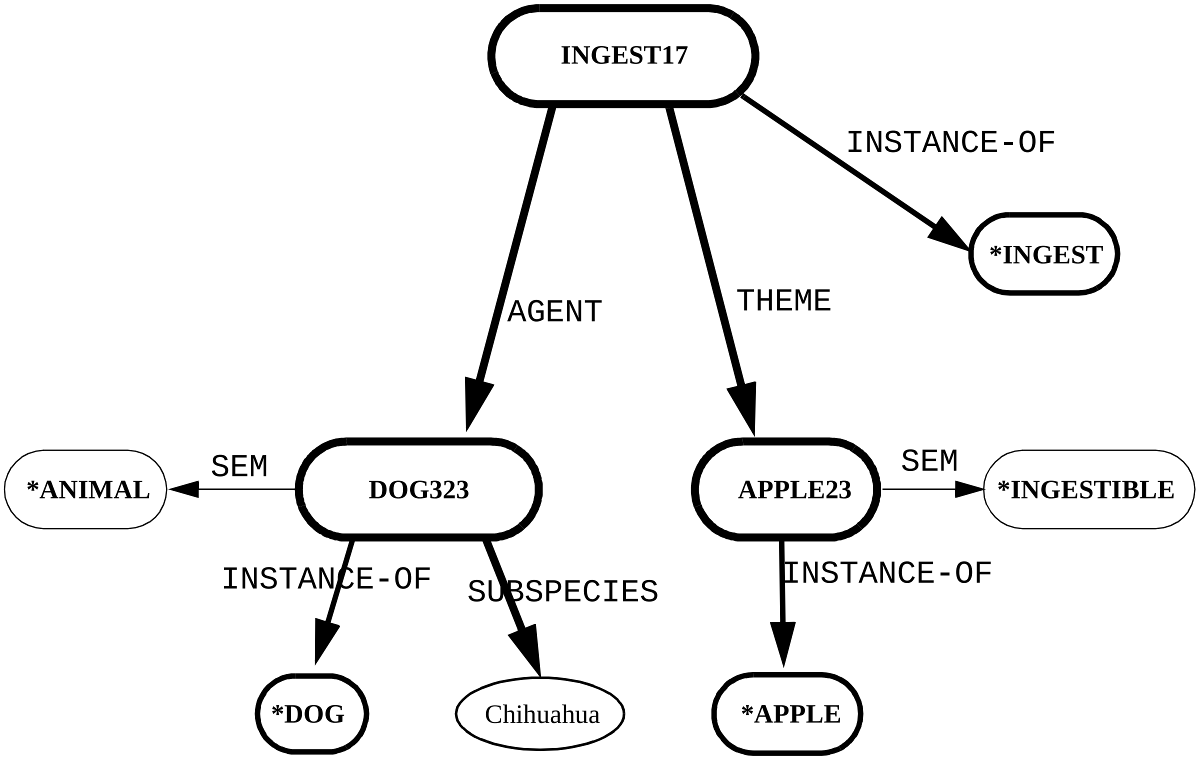


Figure 5C. Graphical representation of initial TMR (one network of structures

```
(DOG323
          (INSTANCE-OF *DOG)
          (SUBSPECIES “CHIHUAHUA”)))
(APPLE23
          (INSTANCE-OF *APPLE))
(INGEST17
          (INSTANCE-OF *INGEST)
          (AGENT    (VALUE DOG323)
                    (SEM *ANIMAL))
          (THEME    (VALUE APPLE23)
                    (SEM *INGESTIBLE)))
```

If more properties of a concept were known, for example if we knew that its color was white, there would be another slot in the DOG323 structure called COLOR with a value of WHITE. The graph notation represents pointers as direct links to the node (instance structure).[1]

A structure may participate in multiple other structures, which would be illustrated in the graph by having multiple arrows pointing to a node. For example, if *chihuahua* were modified by the adjective *horrible*, a structure (of type ATTITUDE) would be added to the graph which would point to DOG323 in the same fashion as the pointer from the INGEST17 concept instance. The details of the TMR notation or the illustrative graph are not salient for our current purpose, which is to illustrate how semantic patterns found in lexicon entries are instantiated combined in order to produce the initial TMR.

6. Elements of Lexical Specification

Before proceeding with a more detailed account of the more important zones of the lexicon, we present a brief overview of several of the static and dynamic knowledge sources and representational formalisms that are referenced in various zones of the lexicon.

This section briefly describes the following knowledge sources: the syntactic representation, the model of world knowledge, and the Text Meaning Representation language. The syntactic structure our system produces during analysis is partially composed of syntactic component structures (called *fs-patterns*) contained in the lexicon entries in the **SYN-STRUC** zone; the representation for entire parse trees is described here in Section 6.1. and the formalism for representing local lexical-syntactic information (in **SYN-STRUC**) is covered later in Section 7. The TMR structure itself (used as the interlingua in translation) is built using semantic patterns in the **LEX-MAP** zone of lexicon entries, and it is sketched out here Section 6.3.; the actual representation that is used in **LEX-MAP** is presented further below in Section 8. Since the TMR is grounded in the ontology, Section 6.2 presents an outline of the world model captured by that knowledge source.

6.1 F-structure

The syntactic structure used in our system is a modification of a Lexical-Functional Grammar (LFG) f-structure representation (we will still refer to it as an f-structure). The traditional LFG f-structure is augmented by a ROOT identifier (akin to the labelling of a node in a tree structure); at each level, the ROOT identifier is followed by the word sense identifier (lexeme name) for the relevant word. The representation can be thought of as a list representation of a (possibly recursive) feature structure, where each attribute name is followed by either a symbol value or another (imbedded) f-structure. For example, the f-structure below is the preferred parse of the sentence *The old man ate a doughnut in the shop*.

```
((ROOT +EAT-V1)
 (MOOD DECL) (VOICE ACTIVE) (NUMBER S3)
 (CAT V) (TENSE PAST) (FORM FINITE)
 (SUBJ ((ROOT +MAN-N1)
        (NUMBER S3) (CAT N)
        (PROPER -) (COUNT +) (CASE NOM)
        (DET ((ROOT +THE-DET1) (CAT DET)))
```

[1] Note that some liberties were taken with the VALUE facet in the text structure; in the graph, the filler of the VALUE facet is represented by the node itself.

```
       (MODS ((ROOT +OLD-ADJ1) (CAT ADJ)
              (ATTRIBUTIVE +))))
(OBJ ((ROOT +DONUT-N1)
      (NUMBER S3) (CAT N) (PROPER -) (COUNT +)
      (DET ((ROOT +A-DET1) (CAT DET)))))
(PP-ADJUNCT ((ROOT +IN-PREP1)
             (CAT PREP)
             (OBJ ((ROOT +SHOP-N1)
                   (NUMBER S3) (CAT N)
                   (PROPER -) (COUNT +)
                   (DET ((ROOT +THE-DET1)
                         (CAT DET))))))))
```

The same information can also be represented as the typed feature structure matrix shown in Figure 6A.

6.2 World Knowledge

The semantic zone of a lexeme and the meaning representation of a text (the TMR) are each defined in appropriate specification languages with their own syntax and semantics. In order for a semantic specification to have explanatory power, the *atoms* of the meaning representation language must be interpreted in terms of an independently motivated model of the world (i.e., our ontology). Our approach to semantics shares this tenet with logical semantic theories (e.g., Kamp's DRT (Kamp 1981)). A major point of difference between these philosophies is the following corollary which logical semanticists do not find compelling: for any realistic experiments to be performed with an NLP system using the algorithms and formalisms suggested by a semantic theory, this world model must be actually built, not just defined algebraically. The issue of grounding symbols has been widely debated in AI, linguistics, philosophy of language and cognitive science (e.g., McDermott, 1978). While not addressing this problem directly in this paper, we would like to point out another well-known position on this issue which is different from ours. Adherents of that approach attempt to ground the semantics of a language in the language itself, by using numbered word senses as atoms in meaning representation and thus equating the object language and the metalanguage of description (e.g., Farwell *et al.* 1993). In some cases, this is augmented with a small number of special predicates (e.g., Jackendoff, 1983, 1990). The resulting semantic descriptions are language-dependent, which necessitates extra work in building multilingual applications (a good example is the work of Dorr and her colleagues in which language-dependent semantic specifications have to be modified in a variety of ways to support a translation application, which, though claimed to be interlingual, is in spirit rather transfer-oriented). Nirenburg and Levin, (1992) and Levin and Nirenburg, (1994) call this approach to semantics *syntax-driven*, while the semantics advocated in this paper is called *ontology-driven*.

The term *ontology* is used here to denote a body of knowledge about the world. Our ontologies (see Carlson and Nirenburg, (1990) for an earlier exposition) are structured as directed graphs, or, more specifically, tangled trees. The knowledge in the world model is separated into two (interconnected) knowledge bases. The first knowledge base, referred to as *ontology* proper, contains knowledge about *concepts*. The second knowledge base, called the *onomasticon*, is a collection of specific instantiations of ontological concepts "remembered" by the system. Thus, the concepts "U.S. President" or "automobile manufacturer" may be found in the ontology, while the

$${}_{\text{+EAT-V1}}\begin{bmatrix}
MOOD & DECL \\
VOICE & ACTIVE \\
NUMBER & S3 \\
CAT & V \\
TENSE & PAST \\
FORM & FINITE \\
SUBJ & {}_{\text{+MAN-N1}}\begin{bmatrix}
NUMBER & S3 \\
CAT & N \\
PROPER & - \\
COUNT & + \\
CASE & NOM \\
DET & {}_{\text{+THE-DET1}}\begin{bmatrix} CAT\ DET \end{bmatrix} \\
MODS & {}_{\text{+OLD-ADJ1}}\begin{bmatrix} CAT & ADJ \\ ATTRIBUTIVE & + \end{bmatrix}
\end{bmatrix} \\
OBJ & {}_{\text{+DONUT-N1}}\begin{bmatrix}
CAT & N \\
NUMBER & S3 \\
PROPER & - \\
COUNT & + \\
DET & {}_{\text{+A-DET1}}\begin{bmatrix} CAT\ DET \end{bmatrix}
\end{bmatrix} \\
PP-ADJUNCT & {}_{\text{+IN-PREP1}}\begin{bmatrix}
CAT & PREP \\
OBJ & {}_{\text{+SHOP-N1}}\begin{bmatrix}
CAT & N \\
NUMBER & S3 \\
PROPER & - \\
COUNT & + \\
DET & {}_{\text{+THE-DET1}}\begin{bmatrix} CAT\ DET \end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}$$

Figure 6A. Matrix f-structure representation

knowledge that the system may have about Harry Truman or Saab will be found in the onomasticon.

6.2.1 The Ontology

The concepts in the ontological world model include *objects* (such as airplanes, ideas, or giraffes), *events* (such as buying or eating) and *properties* (such as *has-as-part* or *temperature*). The ontology is organized as a tangled taxonomy (an `IS-A` hierarchy) for reasons of storage and access efficiency. Thus, the concept `HAMMER` may be a child (i.e., a specialization) of the concept of `HAND_TOOL`, while concepts of `BALL_PEEN_HAMMER` and `CLAW_HAMMER` could be located under `HAMMER` (`CLAW_HAMMER IS-A HAMMER IS-A HAND_TOOL`). Ontological entities could also be understood as the perception of Platonic ideals or natural kinds, as represented in the world model. In other words, the HAMMER concept does not refer to a particular hammer, but to the generic notion of a hammer. Ontological concepts can be *instantiated*, that is, a representation of a specific instance of the concept is produced to signify a particular mention of this concept in a text. Thus, contract-132 may refer to the contract referred to in the seventh sentence of the text that a semantic analyzer is processing at the moment.

In addition to the organization into a taxonomy via IS-A links, the ontology also contains numerous other links between concepts; a link (other than IS-A) between two concepts is essentially a *property* of the source concept, which has, as the value, a pointer to the destination of the link. These additional properties are used as background knowledge for building and disambiguating semantic dependency structures in TMRs. Figure 6B illustrates a fragment of a hypothetical ontology, with mostly taxonomic (IS-A) links shown. An ontology that will actually be used in an application will include such properties as, for instance, IS-PART-OF, IS-AN-OCCUPANT-OF, MANUFACTURED-BY as well as semantic dependency relations that have been traditionally referred to as *case roles* in the Case Grammar and its many practical applications. In our system, we represent ontological concepts as *frames*, while properties are represented as *slots* in the FRAMEKIT or FRAMEPAC frame representation languages. Graphically, concepts are represented as nodes and properties as labelled links between nodes. For example, the EAT concept may have case role slots such as AGENT and THEME (reflecting the eater and what is being eaten), as well as slots that are more general, such as LOCATION (probably *inherited* from an ancestor of EAT in the ontology and not directly acquired for the concept EAT).

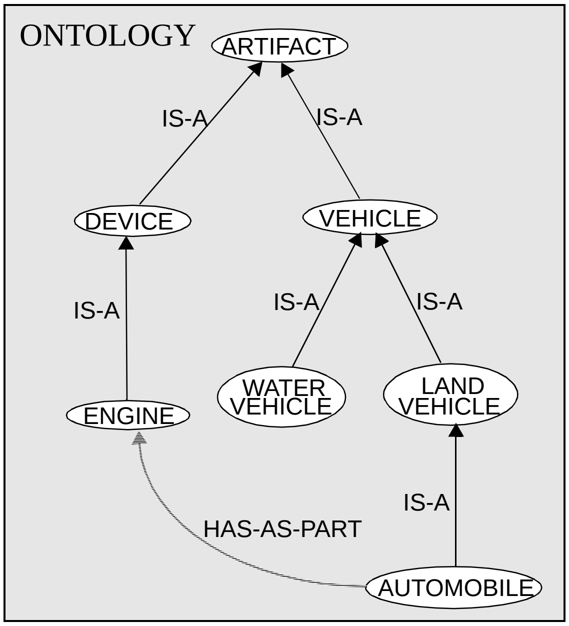


Figure 6B. An ontology fragment

All the above properties are, in fact, relations between ontological concepts. Another kind of property in our system is called *attribute* and signifies a link between a concept and a specially defined set of values (numerical, literal or scalar). Properties like ENGINE_TYPE or TEMPERATURE are attributes. Properties are defined in frames for particular concepts and, in accordance with the semantics of the representation language, apply to all concepts below them in the hierarchy. Constraints are placed in the definition of a property on domain and its range; these constraints are also concepts from the ontology. When the property appears as a particular slot in the frame for a concept, additional semantic constraints may be locally defined in this frame. These will be more specific than the constraints already specified in the definition of the property. For example, there might be a `LOCATION` relation defined in the ontology. The domain of this relation might be specified as any `EVENT` or any `PHYSICAL_OBJECT` (in other words, events and physical objects may have locations). The range of the relation might be `PLACE` (that is, only places can be the locations of events or physical objects). The concept of an `AIRPLANE_LANDING_EVENT` would have a `LOCATION` slot (being, presumably, a descendent of `EVENT`, this concept is within the domain of the relation). However, it may be useful to further constrain the range of the relation (i.e., the allowed value of the slot) in this particular concept to be, say, `LANDING_STRIP`, a descendent of `PLACE`. This further constraint may be overridden in some text occurrences (as in texts about forced or crash landings), and the algorithm discussed in Section 10 incorporates a constraint relaxation technique to take care of such situations. In FRAMEKIT and FRAMEPAC, the constraints on the allowed fillers of various slots are maintained in the `SEM` facet of the slot, whereas the fillers themselves are in the `VALUE` facet.

6.2.2 The Onomasticon

The onomasticon contains instances of ontological concepts that are *remembered* by the system (persistent).[1] The coverage of this type of a knowledge base can be different in different applications. For instance, it can contain the facts that a speaker assumes to share with a particular hearer, to facilitate pragmatically appropriate dialogue. Thus in addition to some core general knowledge about named instances in the world (such as Ford Motor Corporation, Richard M. Nixon, and Ottawa), the onomasticon may be augmented with domain-specific knowledge of remembered instances that might be necessary for a particular application (perhaps including regional detailed gazetteer information, names or persons or companies of interest, dates or events of interest, etc).

Concept instances in the onomasticon can be named (that is, reflect proper noun names such as Richard M. Nixon or Ottawa), and may be referred to as *named instances*. Examples of named instances include geographical and political entities or names of people. These names would, of course, be in a particular language, which would sometimes necessitate special treatment in machine translation applications.

In addition to instantiations of entities, it may also be useful to encode, in a knowledge base, instantiations of events. The Battle of Gettysburg may be such an event that could be useful for some domains, and hence may be included in the static knowledge base for a particular application or domain.

Knowledge bases of instantiations of concepts may be either static or dynamic. Instantiations of countries and cities, for example, would fall into a static knowledge base, because this type of information would be obtained from gazetteers or from similar references. Instantiations may also be of a more dynamic nature, along the lines of what used to be called *episodic memory* in cognitive science, for example, in Tulving (1985).

[1] The Oxford English Dictionary defines onomasticon as “a vocabulary or alphabetic list of proper nouns, esp. of persons. Formerly used more widely of a vocabulary of names, or even of a general lexicon”

Onomasticon entries are referenced from the lexicon of any language. Thus, for a given language there would be lexicon entries for Japan, Paris, and John F. Kennedy, pointing to the appropriate instantiated concept in the onomasticon, and with the appropriate name for that language forming the citation form.

6.3 Text Meaning Representation Language

Since the lexical semantic specification in our lexicon (i.e., the **LEX-MAP** field in the lexicon) is in terms of underspecified TMR fragments, a sketch of the TMR language is needed before proceeding to a discussion of the lexical semantic specification itself (Section 8 below). As stated above, the TMRs are built, in part, by combining instantiations of **LEX-MAP**s from the words in the sentences. Further discussions of TMR can be found in Carlson *et al* (1994) and Nirenburg and Defrise (1991).

We stated the goal of computational semantics as capturing the meaning of input text in an unambiguous machine-tractable representation; this section introduces the TMR formalism, in which that unambiguous machine-tractable representation of meaning is rendered. A TMR expression captures the explicit and some of the implicit information of a natural language text. In addition to the basic semantic content, TMR sets out to capture pragmatic factors, including focus, textual relations, speaker attitudes, and stylistics.

We use the acronym TMR to refer both to the language and to the rendering of the meaning of a text, utilizing the TMR language; the differences should be absolutely clear from context. A TMR is implemented as a network of typed FRAMEKIT or FRAMEPAC frames. Frame types include instantiated ontological concepts (often with additional properties listed), speech acts, relations (e.g., *causal*) among frames in the network, and speaker attitude frames. The sections below discuss each of these TMR constructs in turn.

6.3.1 Propositional Content

Basic semantic content or meaning of an utterance (sometimes called the propositional content or the “who did what to whom” component) is represented in a TMR representation as a network of instantiated concepts from the ontology (or imported instances from the onomasticon), combined and constrained in various ways. The semantic analysis processes (see Section 10) crucially rely on lexical-semantic information (as defined in Section 8) from the appropriate lexicon entries. To obtain the semantic content of complex structures with dependencies, information about the argument structure of lexical units (also stored in the lexicon) is used. This information relates not only to ambiguity resolution in “regular” compositional semantics (i.e., straightforward monotonic dependency structure building), but also the identification of idioms, treatment of metonymy and metaphor, and resolution of reference ambiguities.

In general, each instantiated concept in a TMR reflects an individual (e.g., thing or event) in the world or in the speakers’ discourse model; however, when considering the TMRs for entire texts, this characterization must be amended to refer to *mentions* of individuals. Thus when a particular individual is referred to in various portions of the text, multiple instantiations reflect the multiple mentions; an explicit set of coreference structures track the relationship among those instantiations. The motivation for this approach (vs. referring to the same instantiation throughout the text) reflects the fact that as a text progresses, new attitudes or properties may become known about the individual; but in generation, it is appropriate to make these new properties known not all at once, at the first mention, but at the appropriate point in the text (i.e., mirroring the source). However, the coreferences do make the cumulative information available, if necessary, for lexical selection or morphology (e.g., gender) in generation.

The meaning of a text cannot be reduced to propositional content alone; what is additionally needed is the representation of pragmatic and discourse-related aspects of language, that is, speech acts, deictic references, speaker attitudes and intentions, relations among text units, the prior context, the physical context, etc. As most of the knowledge underlying realizations of these phenomena is not society-general, universal, or constant but is rather dependent on a particular cognitive agent (a particular speaker/hearer) in a particular speech situation and context, the pragmatic and discourse knowledge units are not included in the ontology (which is supposed to reflect, with some variability, a relatively static model of the world). The representation of this "meta-ontological" information is thus added to the representation of meaning proper to yield a representation of text meaning.

Most of the non-propositional components of text meaning are also derived from lexical clues in the input (see example lexicon entries below and Section 8.3). Some of the most important non-propositional components of the TMR representation formalism are reviewed below (for more detailed discussion see Nirenburg and Defrise (1991)), specifically speaker attitudes, stylistic features, and rhetorical relations.

6.3.2 Attitudes and Modality

A critical aspect of capturing the intent of a speaker in a meaning representation is rendering the *attitudes* that the speaker holds toward the objects or situations which are represented in the propositional (ontology-based) component of text meaning representation. The speaker may also convey attitudes about the speech act in which the utterance was produced, about elements of the speech context, or even about other attitudes. Similarly, the speaker may convey events, certain relations (and sometimes other constructions) in a particular *modality.*

These attitudes and modalities are conveyed in TMR by a quintuple (either an `attitude` or a `modality`) consisting of a `type`, a `value` in the interval *[0, 1]*, an `attributed-to` slot (identifying the person who holds the attitude, typically the speaker), a `scope` (identifying the entity towards which the attitude is expressed or the event etc. for which the modality is expressed), and a `time` (representing the absolute time at which the attitude was held). As is the case with all TMR constructs, attitudes and modalities may be either lexically triggered (i.e., explicitly specified in the **`LEX-MAP`** of a lexeme, such as the word *doubt* or *could*) or triggered by other, non-lexical phenomena, such as syntax or morphology (for example, by a diminutive form).

The following attitudes and modalities are among those used in the TMR language (for present purposes, the distinction between attitudes and modalities isn't relevant):

- `Epistemic`, ranging from *speaker does not believe that X* to *speaker believes that X*.
- `Evaluative`, ranging from *worst for the speaker* to *best for the speaker.*
- `Deontic`, ranging from *speaker believes that the possessor of the attitude must do X* to *speaker believes that the possessor of the attitude does not have to do X.*
- `Potential`, ranging from *the possessor of the attitude believes that X is not possible* to *the possessor of the attitude expects that X is possible*

- `Volitive`, ranging from *the possessor of the attitude does not desire that X* to *the possessor of the attitude desires that X*.
- `Salient`, ranging from *unimportant* to *very important*. This varies with the importance the speaker attaches to a text component, thus has some overlap with the notion of focus.

6.3.3 Stylistics

Even when the stylistic overtones or nuances of a lexical entry do not contribute directly to the propositional semantics of a text, they can still convey some element of meaning, whether it be in conveying attitudes, setting a mood, or using rhetorical devices such as irony. Thus we identify that the stylistics of a lexeme needs to be encoded in a lexicon entry, in addition to the lexical semantic information. In encoding lexicons for languages with rich social deictics, such as Japanese, the issue of stylistics becomes even more acute.

The TMR representation includes a set of style indicators which is a modification of the set of *pragmatic goals* from Hovy (1988). This set consists of six stylistic indicators: *formality, simplicity, color, force, directness,* and *respect*. In order to obtain this resulting TMR stylistic representation (essentially a set of overall values for the entire utterance, or multiple sets scoping over substrings of the utterance) it is necessary to label various lexical entries (including idioms, collocations, conventional utterances, etc.) with appropriate representations of values for these stylistic indicators. Values for these factors are represented as in the interval *[0,1]*, where *0.0* is low, *1.0* is high, and *0.5* represents a default, neutral value. In the semantic analysis process, the values are available for assisting in disambiguation (relying on expected values for the factors and utilizing the heuristic that typically the stylistics will be consistent across words in an utterance). The resulting values in the TMR representation help guide generation, etc.

Some examples of English lexical entries that might include style features are:

```
upside:   formality - low
          color - high
delicious:formality - somewhat high
          color - high
great:    formality - low
one (pronominal sense): formality - high
          force - low
```

6.3.4 Relations

Relational structures are used in TMRs to capture the relationships and connections between structures in the TMR, between real-world entities, elements of text, etc. TMRs include the following inventory of relations: Domain Relations, which represent real-world connections (and, therefore, are instantiations of ontological relations) between objects or events; Textual Relations such as rhetorical relations, e.g., conjunction and contrast, refer to properties of text itself; Temporal Relations, expressing a partial ordering between `TIME` structures; Coreference Relations identify that two instantiations in fact refer to the same real-world entity (although, possibly, at different time intervals); and Quantifier Relations, which are used for relations between numerical quantities.

Since domain relations play such a salient role in the TMR in linking together events and/or objects, we sketch an inventory of these relations. Domain relation types in a recent version of the TMR specification include the list below; this list is not meant to be definitive or exhaustive, but a reflection of the list in progress, mainly driven by empirical needs of creating TMR representations in a few languages.

- `CAUSAL` Domain Relations. Relations of dependence among events, states, and objects; can be either Volitional (the relation between a deliberate, intentional action of an intelligent agent, and its consequence) or Non-volitional (the relation between a non-intentional action or a state of an intelligent agent and its consequence.. Subtypes: `Reason`, `Enablement`, `Purpose`, `Condition`
- `CONJUNCTION` Domain Relations. Relations among adjacent elements that are components of a larger textual element. Subtypes: `Addition`, `Enumeration`, `Contrast`, `Concession`, `Comparison`
- `PARTICULAR/REPRESENTATIVE` Domain Relations. Relations which identify that one element is an example, or a special case, of the other element. Subtypes: `Particular`, `Representative`
- `ALTERNATION` Domain Relations. Relations that are used in situations of choice, parallel to the logical connector “OR.” Subtypes: `Inclusive-or`, `Exclusive-or`
- `COREFERENCE` Domain Relation. The relation established among textual references to an object, an event, or a state.
- `TEMPORAL` Domain Relations. Identify when one event (or object instance/snapshot) happened relative to another. Subtypes: `At`, `After`, `During`

Recent inventories of relations, such as those in Hovy *et al*. (1992) and Hovy and Maier (forthcoming) may allow us to restructure and complete our domain (and textual) relation inventory; our current inventory has been developed as necessary, based on examples from our corpora.

7. Lexical Syntactic Specification

The contents of the **`SYN-STRUC`** zone of a lexicon entry is an indication of how the lexeme fits into parses of sentences. In addition, this zone provides the basis of the syntax-semantics interface. Thus a brief discussion of this zone is necessary to understand the semantic analysis process (briefly described in Section 10), which relies on the syntax-semantics interface as the main dynamic knowledge source used in the process of constructing a semantic representation (i.e., the TMR) from input text.

The information contained in the **`SYN-STRUC`** zone of a lexeme is essentially an underspecified piece of a parse tree (f-structure) of a typical sentence (as specified in Section 6.1); this underspecified piece, called an *fs-pattern*, contains the lexeme in question, and may include information from one or two levels of structure above and/or below the current lexeme. The fs-patterns of all the words, morphemes, and syntactic constructs unify to form the f-structure parse of the sentence (although, obviously, the search process involved in syntactic parsing takes a more circuitous route).

Since f-structures do not indicate linear order, the fs-pattern is essentially a dependency or immediate dominance structure. In the simple case, the fs-pattern for a verb will indicate the arguments for which the verb subcategorizes. In LFG f-structures, all arguments (including subjects) are immediate children of the verb node, so the selection in the fs-pattern is for elements which are descendants of the current lexeme in the f-structure tree. We use the same mechanism for syntactic relationships other than arguments. So adjectives and prepositions, for example, select (in their respective fs-patterns) for the syntactic head which they modify (in addition, prepositions select for their arguments.)

In the fs-patterns, we place variables at the ROOT positions selected for by the lexeme in question, which is identified by the variable $var0; this allows the fs-patterns to be inherited (using the **SYN-S-CLASS** mechanism described below). Subsequently numbered variables ($var1, $var2, ...) identify other nodes in the f-structure with which the current lexeme has syntactic or semantic dependencies. For example, the fs-pattern below is appropriate for any regular monotransitive verb:

```
((root $var0)
 (subj ((root $var1) (cat n)))
 (obj ((root $var2) (cat n))))
```

Or, viewed as a feature structure:

$$\boxed{0}\begin{bmatrix} \text{SUBJ} & \boxed{1}\begin{bmatrix} CAT\ n \end{bmatrix} \\ \text{OBJ} & \boxed{2}\begin{bmatrix} CAT\ n \end{bmatrix} \end{bmatrix}$$

The exact syntactic relationship of words in a sentence may vary due to syntactic transformations, valency changes, or movement rules; the variables support a level of indirection in the fs-patterns. Additional advantages of this mechanism include the ability to inherit fs-patterns from a hierarchy, as well as reducing the work in assigning correspondences between lexical functions and case roles.

In cases of lexicon entries for idioms, verbs with particles, non-compositional collocations, etc., the ROOT attribute in an fs-pattern may be followed by a specific lexeme instead of the variable. For example, the special sense of *kick* which defines the idiom *kick the bucket* will select for an OBJect with ROOT **+bucket-n1**, where **+bucket-n1** is a lexeme identifier for a standard sense of the word *bucket*. Additionally, in the fs-pattern, the attribute-value pair will be followed by the symbol null-sem as follows: (ROOT +bucket-n1 null-sem) to indicate that this word sense does not contribute to the semantics of the phrase. In cases of of idioms such as *spill the beans*, *spill* will select for an OBJect which will specify (ROOT +beans-n3), meaning that this special sense of *beans* (meaning *information*) does contribute its meaning as an idiom chunk to the entire idiom. In both of these cases it is obligatory to specify the root, so the special sense in question will fail the syntactic parse (in analysis) if the selected root does not appear in the utterance. In generation, any special sense will get selected in the lexical selection process only if the meaning is appropriate.

The **SYN-STRUC** zone has two facets. If the word is syntactically regular (that is, non-idiomatic, has no particles, etc.), then the **SYN-S-CLASS** facet is used to indicate which fs-pattern to inherit from the class hierarchy of fs-patterns (see, e.g., Mitamura, (1990) for an early description of this kind of mechanism). If none of the class fs-patterns are appropriate for the lexeme in question, an fs-pattern may be locally specified in the **SYN-S-LOCAL** facet; in fact, both a class and local information may be specified, and the two fs-patterns are unified.

In addition to specifying syntactic dependency structure, the fs-pattern also indicates an interaction with the meaning pattern from the **SEM-STRUC** zone. Certain portions of the meaning pattern for a phrase or clause are regularly and compositionally determined by the semantics of the components (Principle of Compositionality); the structure of the resulting meaning pattern is determined not only by the semantic meaning patterns of each of the components, but also by their syntactic relationship in the f-structure.

8. Lexical Semantic Specification

The lexical semantics of a lexical unit is typically represented in the **LEX-MAP** field of the **SEM-STRUC** zone of a lexical entry. In the simplest case, the **LEX-MAP** links the lexical unit with an ontological concept; thus, the essence of the lexical meaning is referring to an ontological concept. Viewed procedurally, the link in the **LEX-MAP** field is an instruction to the semantic analyzer to add an instance of the ontological concept in question to the nascent TMR. So, for example, one sense of the English word *dog* might be treated in our system as a link to the concept DOG in the ontology, or, in other words, a command to create an instance of it (e.g., DOG34). The meaning assignment mechanism works this simply only in the case of one-to-one mapping between word senses and ontological concepts, which is not necessarily the case for many lexical units, as is discussed below. More complex mappings are required for most lexical units in a realistic lexicon.

As is discussed in greater detail in Section 9, there is a spectrum of possible divisions between putting information into the ontology vs. putting information into the lexicon. At one extreme, each lexeme maps into exactly one concept from the ontology, and, at the other, a meaning can be expressed as an arbitrarily complex combination of ontological concepts (primitives). We choose a position which is different from both the extremes, for the following reasons.

While it may often be possible to express the meaning of a lexical unit as a link to a single ontological concept, in a large portion of cases, this would run into problems discussed in the sections below. In general, though, meaning mappings are made to concepts which are “closest” to the meaning of the lexeme while still remaining more general than the latter. The link to such a concept is recorded in the **LEX-MAP** but then additional constraints are added, so that when this lexicon entry is actually used, the instantiated concept would include these additional lexicon-specified constraints. These additional constraints can either add information to the concept as it is specified in the ontology, override certain constraints (e.g., a selectional restriction), or indicate relationships with other concepts expected in the sentence. The sections below deal with these two basic cases of lexical-semantic mapping in our system: simple one-to-one mapping, which we call *univocal* mapping, and then complex, or *constrained* mapping.

8.1 Univocal Mapping

The univocal mapping of exactly one concept to lexeme is utilized when the concept denoted by the lexeme is rather *universal* (essentially, universal has come to mean 'common to the languages we, or our informants, know'). As additional languages are treated and cross-cultural concepts come to be reflected in the ontology, the share of univocal entries may increase or decrease. Examples of universal concepts might include the meaning of **+die-v1** (in the most literal sense of

’cease to live’), natural kinds such as *tree*, *dog*, artifacts or terms in technical sublanguages, etc. Clearly, when constructing a practical lexicon and ontology, these universal concepts are derived somewhat intuitively, and may reflect the pragmatics of the textual domain in question; for example, technical domains tend to have a high percentage of such concepts.

Our notation for a univocal lexical semantic mapping is straightforward. Thus, the primary sense of the word *dog*: **+dog-n1** will have the following **SEM-STRUC** zone:

```
SEM-STRUC:
        LEX-MAP:
                (%dog)
```

“%″ indicates an ontological concept that is to be instantiated when the meaning in question is included in the overall semantic dependency structure of a sentence in which *dog* appears. Note that the name of the concept from the ontology (or onomasticon) need not be the same as that of the lexeme in question. A univocal mapping between lexeme and ontological concept merely implies that all constraints on an ontological concept (i.e., all information provided within the frame for that concept in the ontology) are consistent with the meaning of the lexeme; we refer to the situation where all the constraints are not consistent as constrained mappings.

8.2 Constrained Mapping

Once the “closest” concept is determined, constraints and further information (including possible reference to other concepts from the ontology) are recorded in the appropriate slots in the lexicon entry. The facet facility of FRAMEKIT and FRAMEPAC is invoked to encode constraints on concepts in a constrained mapping of a semantic specification. The constraining or “specialization” facets used in the lexicon representation are as follows:

- VALUE - a specific value (e.g., number of sides for a triangle = 3, sex of a man = male). This is the facet where actual information is represented; typically, the other facets are constraints on what may be a legal (or likely) filler of the VALUE facet. Typically, in the ontology, this facet is not specified. This facet is used for recording a) constrained mappings within lexical semantic specification, or b) semantic dependency structure links. Fillers of this facet are often symbols consisting of “^” appended to a variable name, e.g.,`(%visit (AGENT (VALUE ^$var1))...)` The caret is an operator (akin to an intension operator) which dereferences the variable (retrieves the lexeme to which the variable gets bound during the syntactic parsing process within the f-structure) and then retrieves the concepts which are instantiated by that lexeme's **LEX-MAP** specification. So any place where a ^$var# appears is an indication to the semantic dependency-building algorithm of how to attempt to build the sentential TMR (see Section 10). In simple terms, ^$var1 means “the meaning of the syntactic unit referenced by $var1.”
- DEFAULT - typical, expected value (e.g., color of diapers = white). If a VALUE is needed by some inference process operating on a TMR representation, and the VALUE is unspecified, the DEFAULT is used; this usage is consistent with standard Artificial Intelligence and logic default mechanisms.
- SEM - akin to a traditional selectional restriction (e.g., the color of a car has to be a COLOR). This is essentially a constraint on what the VALUE may be. Instead of using some small set of binary features, we allow any concept (or boolean combination of concepts) from

the ontology to be a semantic constraint; any VALUE then needs to be a descendent of one of the concepts listed in SEM. All slots have SEM facets in the ontology, but often these need to be modified (typically, constrained further) for a specific lexeme. This semantic restriction is not absolute; it may be relaxed or violated in specific ways in cases of metonymy or metaphor.

- RELAXABLE-TO - maximum relaxability, if any, of SEM restrictions; used in cases of selectional restriction violation processing (treatment of unexpected input, including metonymy and metaphor).
- SALIENCE - a scalar value in the range *[0.0, 1.0]* designating the significance of a specific attribute slot or role (partly reflecting the notion of “defining properties” vs. “incidental properties”).

The following example illustrates a simple case of lexical semantic mapping for the lexeme **+eat-v1**. The **SYN-STRUC** lexicon zone contains the lexical syntactic specification of the lexical entry, in which the subcategorization pattern of the verb is described:

```
SYN-STRUC:
     SYN-S-LOCAL:                ; (for example only  – should be CLASS)
          ((root $var0)          ;$var0 gets bound to +eat-v1
           (subj ((root $var1);$var1 gets bound to head lexeme
                  (CAT n)))      ;whose lexical category is N
           (obj ((root $var2)  ;$var2 gets bound to head lexeme
                 (CAT n))))))  ;this is also a noun phrase
```

During analysis, the variables $var1 and $var2 are initially bound to “placeholders” for the lexical semantics of the subject and object of the verb, respectively. Once the lexical semantics of those syntactic roles is determined, the semantic composition process gets under way. If this process is successful, a semantic representation (the TMR) for a higher-level text component is produced. The **SEM-STRUC** zone of the lexicon entry for **+eat-v1** contains linking information as well as selectional restrictions, constraints on the properties of the meanings of the verb’s syntactic arguments:

```
SEM-STRUC:
     LEX-MAP:
          (%ingest                   ;+eat-v1  maps into %ingest
             (AGENT (VALUE ^$var1)    ;subject maps into agent
                    (SEM *animal))
          ;the meaning should be a descendent
          ; of ontological concept *animal
             (THEME (VALUE ^$var2)     ;object maps into theme
                    (SEM *ingestible)
                    (RELAXABLE-TO *physical-object))))))
          ;theme’s meaning should be a descendent of  *ingestible
          ;or at least of a *physical-object
```

This structure can also be represented as a feature structure matrix:

$$\begin{bmatrix} AGENT & \begin{bmatrix} VALUE & \text{\textasciicircum}\boxed{1} \\ SEM & \text{*animal} \end{bmatrix} \\ THEME & \begin{bmatrix} VALUE & \text{\textasciicircum}\boxed{2} \\ SEM & \text{*ingestible} \\ RELAXABLE - TO & \text{*physical-object} \end{bmatrix} \end{bmatrix}_{\text{ingest}}$$

Traditionally, selectional restrictions are defined in terms of a small fixed set of concepts or features; we have found that it is often useful to use arbitrary concepts from the ontology as “selectional restrictions”. These constraints are represented in the SEM facet, and can be arbitrary concepts from the ontology:

+taxi-v1 (sense of ‘move-on-surface’, said only of aircraft, e.g., *The plane taxied to the terminal*; *The hydroplane taxied to the end of the lake*)

```
SEM-STRUC:
     LEX-MAP:
          (%move-on-surface
            (THEME
              (SEM *aircraft)    ;;the SEM facet
              (RELAXABLE-TO *vehicle))))
```

Note that there may also be “second-order” constraints (i.e., constraints on constraints):

+jet-v1 (literal sense of 'to travel by jet', e.g., *The presidential candidate spent most of the year jetting across the country from one campaign rally to another*

```
SEM-STRUC:
     LEX-MAP:
          (%move
            (THEME
               (SEM *aircraft
                    (PROPELLED-BY
                         (VALUE %jet-engine)))))
```

Thus we see that semantic constraints in this approach can be any arbitrary concept, constrained concept, or even set of concepts from the ontology; this substantially extends the traditional notion of selectional restriction to more fully utilize the knowledge available (from the ontology) for disambiguation.

It is not expected that the meaning of verbs will always be a link to an ontological concept of type EVENT; or that meanings of nouns will uniformly be descendents of the ontological concept OBJECT. There is a great deal of variance in the correspondences between ontological subtrees and parts of speech. For example, many adjectives and nouns (such as *abusive* or *destruction* in English) may be represented as events, whereas many verbs map to attitudes or properties (e.g., *own* or *reek*).

8.3 Non-Propositional Mapping

In addition to the direct or modified mapping into ontological concepts as outlined above, three other scenarios can occur in lexical semantic definitions (represented in **SEM-STRUC** zones of corresponding lexicon entries), either in conjunction with a propositional mapping (and/or each other) or without such a mapping.

The first case involves situations where the meaning of a lexeme corresponds not to a concept, but to a particular filler of a slot defined in another concept; for example, the basic attributive sense of the adjective *hot* maps to a particular value of the TEMPERATURE slot (property) of the meaning of the noun it modifies. In some cases, the semantics of the lexeme indicate the name of the property which connects the meaning of two other lexemes. For example, the locative sense of *in* suggests LOCATION as the property on which the meanings of the prepositional object and its attachment point are linked; thus, the meaning of *in* in the phrase *the dog in the park* is that the meaning of *the park* fills the LOCATION slot of the meaning of *the dog*. Many syntactic morphemes (including many case markings) exhibit this kind of semantic behavior.

The second case involves mapping to TMR constructs which are non-propositional, hence non-ontological, in nature — speaker attitudes, stylistic factors, etc. The representation of this "para-ontological" information (also in the **LEX-MAP**) is in addition to the representation of any propositional meaning; both types of information together form the TMR. As is the case with propositional information, lexical entries may specify which specific constructs those entries trigger as contributions to TMRs of entire texts. The example below illustrates both of the cases mentioned above. The lexical semantics of the lexeme **+delicious-adj1** contains the following two structures:

```
SEM-STRUC:
     LEX-MAP:
          (^$var1
            (instance-of (SEM (value *ingestible))))

          (ATTITUDE
            (type (value evaluative))
            (attitude-value (value 0.8))
            (scope (value ^$var1))
            (attributed-to (value *speaker*))))))
```

The first construct places a semantic constraint (i.e., it must be a descendent of *ingestible) on the meaning of what the adjective modifies (referred to by the variable ^$var1). The evaluative ATTITUDE scopes over this same meaning. The attitude is attributed to the speaker, which is a default value.

The third case involves special treatment, different from the usual instantiation/combination processing. For example, the meaning of the definite article *the* in English, at least in one of its senses, involves reviewing the discourse or deictic contexts for entities of a particular semantic type. The meaning of that sense of the article would not involve the instantiation of any new concepts, but will rather serve as a clue for the identification of previously-instantiated concepts for reference.

9. Lexical Semantics/Ontology Trade-offs

There is no consensus in the semantics or knowledge representation fields about the granularity of the ontology or world model; the granularity decision has a profound impact on the lexical semantics zone in the lexicon. One view of the ontology is to have a one-to-one correspondence between every word-sense in the lexicon and a concept in the ontology. This *word-sense* view of the ontology, in addition to the obvious disadvantage of rampant proliferation of ontological concepts (also called *ontological promiscuity* in Hobbs (1985)), leads to problems in multilingual applications — often roughly comparable words in different languages do not "line up" the same way; this, in turn, leads to further proliferation of new concepts with every new language, as well as inaccurate lexical mappings. These and other problems make this approach impractical for real applications with refined semantic discrimination.

Another well-known approach, which may be called the *decompositional* approach, utilizes a small restricted set of primitives which are combined or constrained in an attempt to render the meaning of any lexical unit. This approach leads to other difficulties in building large-scale world models and capturing shades of meaning; it is not clear that it is possible to derive a set of *a priori* primitives and a compositional formalism which would be expressively adequate to capture the meanings of all desired word senses. Additionally, this approach can yield enormous, unmaintainable lexicon entries for complex concepts.

The approach taken in the model adopted here lies somewhere in between the word-sense and the decompositional approaches, as expressed in Nirenburg and Goodman (1990): *<<<<Need more relevant discussion of how it lies somewhere in between>>>>>>*

> "Viewed as an object, developed in a concrete project, an interlingua should be judged by the quality of the translations that it supports between all the languages for which the corresponding SL-interlingua and interlingua-TL dictionaries have been built. As a process, its success should be judged in terms of the ease with which new concepts can be added to it and existing concepts modified in view of new textual evidence (either from new languages or from those already treated in the system.)" (p. 9).

The notion of "completeness as proof of feasibility for interlinguae" (p. 10) is rejected in the design of the ontology; the ontology is not determined *a priori*, but rather, is updated and revised as new lexemes are entered into the lexicon, as new cross-linguistic evidence of shared concepts arises, and as the domain of the ontology is shifted. The TMR therefore reflects this decision as to the scope of the ontology.

10. Using the Lexicon

This section is an overview of the foundations and methodology of semantic analysis espoused in this work, where semantic analysis is viewed as a component of a knowledge-based machine translation system. As was specified above, the overall goal of this work to capture as much as possible of the meaning of an input text using a set of well-defined structures in an unambiguous machine-tractable knowledge representation language, namely the TMR. In what follows we illustrate how the complex lexical entries (as defined above) are used in the semantic analysis process.

We consider semantic analysis to combine the construction of a basic *semantic dependency structure* (SDS) and augmentation of this structure with additional constraints and other information (such as reference, resolution of deixis, etc.) gleaned from the available lexical, syntactic and other evidence in the input; in our discussion here, the TMR is the representation of this semantic dependency structure. In its most straightforward incarnation, the SDS-building process relies on meanings of atomic lexical units, as defined through links to the ontology and by non-propositional meaning elements; the SDS-building process is guided by the syntax-semantics interface manifested in the lexical syntactic and lexical semantic specifications of lexical entries.

In this section we illustrate the SDS-building process through a simplified example. The readings or semantic interpretations generated by the SDS-building process (both the intermediate and the final results) are expressed in the TMR language. The particulars of this language (defined in Section 6.3 above) are not of critical importance, as long as the language meets a number of criteria regarding its expressiveness and other properties. Regardless of the language, the meaning is represented as an augmented network of instantiations of concepts from the ontology.

The goal of the SDS-building process is to find the most appropriate semantic interpretation of the input text. Each intermediate or final semantic interpretation (as represented by a full or partial TMR) is called a *reading*; many possible candidate readings are constructed and evaluated during the search for the best reading. Candidate readings are ranked according to their *preference* values (the use of this term is different from its familiar meaning introduced by Wilks (1975)). If the assignment of preferences by the search process is appropriate, then the interpretation with the highest preference value at the end of processing should indeed be the one which human translators would choose. Preference values are used by the search heuristic both for pruning paths with low preferences, and for guiding a best-first search method. The preference in the current implementation is a value in the interval *[0.0, 1.0]*, with adjustments to the preference typically made by a multiplier.

Both incrementing and decrementing adjustments are possible, reflecting an increased likelihood on that reading (for example, if the reading reflects the use of a typical collocation or idiom) or a decreased likelihood on that reading (as is the case when any constraint violation occurs). Determining how to adjust preference values in a particular case is an issue of critical importance to the success of this approach, and a variety of factors influence this decision.

The process of building semantic dependency structure is interpreted as traversing a search space of all possible semantic constructions (both well-formed and incomplete) in order to find the semantic construction that best represents the meaning of the input text. Each state in the search space represents one reading, and has an associated preference reflecting the likelihood of that reading. A particular state may be final (i.e., well-formed and complete), or incomplete (where portions of the meaning of the text have not been incorporated into the reading yet). The two operators for expanding or traversing nodes in the search space (i.e., the processes which actually build the SDS) are *instantiation* and *combination*.

The instantiation process *instantiates* each syntactically appropriate word sense from the input syntactic structure (produced by a syntactic parser, the existence of which is assumed) according to the lexical semantic specification of that word sense. Note that the final TMR does not include any syntactic information about the input string. Syntactic information is used as a set of clues necessary (though, certainly, not sufficient!) to guide the semantic analysis process. The syntactic parse identifies the lexemes corresponding to words, idioms, or morphemes in the input string, and eliminates those lexemes which do not meet basic syntactic constraints. The **MORPH**, **CAT**, **SYN**, and **SYN-STRUC** zones of the lexicon entry for each lexeme are utilized during the course of the syntactic parsing process; the **SYN-STRUC** zone provides the most information about the local syntactic context in which the lexeme appears. The syntactic parse structure is used in the application of the *combination* operator, which builds the SDS that forms the bulk of the TMR.

Given the meanings of individual words (recall that these meanings are represented as semantic structures typically corresponding to possibly augmented instantiations of ontological concepts), the combination operator attempts to combine such structures into the meaning of a phrase. For example, the syntactic specification for a sense of *eat* may subcategorize for an object, and the syntax-semantics interface (i.e., the $vars) indicates that the meaning associated with the object serves as the THEME of the meaning of *eat*. Thus the combination operator builds a semantic structure by attempting to insert the meaning representation produced by instantiating the syntactic object into the THEME role in the meaning representation of *eat*. As the combination operator can be applied recursively to phrases, it is expected that, after a series of applications, the meaning of a complete utterance will be produced (in practice, this process in actually implemented in a bottom-up manner). The combination operator is not typically applied at the suprasentential levels of semantic analysis.

Technically, the combination process creates relations between two concepts. These relations are made manifest in the formalism by allowing a slot in one concept have the name of another concept as its value. A number of constraints guides this linking — a) the constraints on the range (and domain) of the relation in the ontology, b) possible further constraints on the relation's range appearing in the head concept's entry in the ontology, and c) the lexical semantics specifying idiosyncratic constraints on the head concept. For example, the case role agent has a constrained range (animals or other animate entities may be agents); the concept ingest constrains the agent to be animal; the German word *freßen* maps to the concept ingest and further constrains the agent to be non-human.

The SDS-building process is also expected to determine which slot will contain a link to the meaning of a dependent element (i.e., the specific relation that holds between the two instantiated meanings) as well as which element is to be the head and which is to be the filler. Three eventualities can be distinguished in this process:

- The syntax-semantics interface explicitly identifies the slot (e.g., the meaning representation of the syntactic subject of *eat* is directed by the content of the lexicon entry for the verb to be inserted into the AGENT slot of the head concept representation).
- An explicit syntactic indicator of the filler's role is available, indicating which element is to be the head, which is to be the filler, and what relation holds between them (e.g., the preposition *in* may indicate that the meaning produced by the object of the preposition fills the LOCATION slot of the meaning produced by the syntactic head to which the prepositional phrase attaches.)
- When no syntactic clue is available as to the nature of the relation between the two elements (in fact, no indication may be available as to which is the head), the SDS-building process undertakes a search over all candidate slots. The head concept of a meaning representation will have a number of allowable ontological properties. The SDS building process includes attempting the slot-filling constraint-satisfaction process over each of these. (This case occurs, for example, in Noun-Noun compounding in English.)

Constraints on slot fillers are defined in terms of ontological concepts. Thus, since the candidate filler is a constrained ontological concept, and the constraint is marked by an ontological concept, too, the constraint satisfaction process can be a matter of verifying that the filler is headed by the concept marking the constraint. In other cases, the constraint satisfaction process involves exploring other (non-taxonomic) paths between the candidate filler and the constraint over the ontology. These other paths may define a metaphorical or metonymic relationship between the candidate filler and the constraining concept.

Given this view of semantic composition as a constraint satisfaction problem, and given also that the candidate filler and the constraints are both ontological concepts, thus, representable as nodes in a connected graph, this process can be interpreted as the problem of finding a low-cost path through a graph, well-known in the graph theory literature. The cost of graph traversal is a function of arc traversal costs, whose relative values are empirically determined. Seeking a low-cost path becomes, then, the control strategy for the process. Conceptually, this process determines an abstract distance between two ontological concepts. The more ontologically related two concepts are, the "cheaper" the path between them. The relation between the two concepts can be vertically taxonomic, or any of a variety of other relations that reflect conceptual relatedness between two concepts (such as composer and his work, *sword* and *scabbard*, part and whole, *to taxi* and airplane, *landing strip* and *airplane*).

Arcs in the graph are directional; the cost of traversing inverse links (and all links have inverse links associated with them) is typically different from the cost of traversing direct links. Graph traversal is typically computed as originating at the candidate filler, and ending at the constraint concept. For example (as illustrated in Figure 10A), the constraining concept for the filler of an

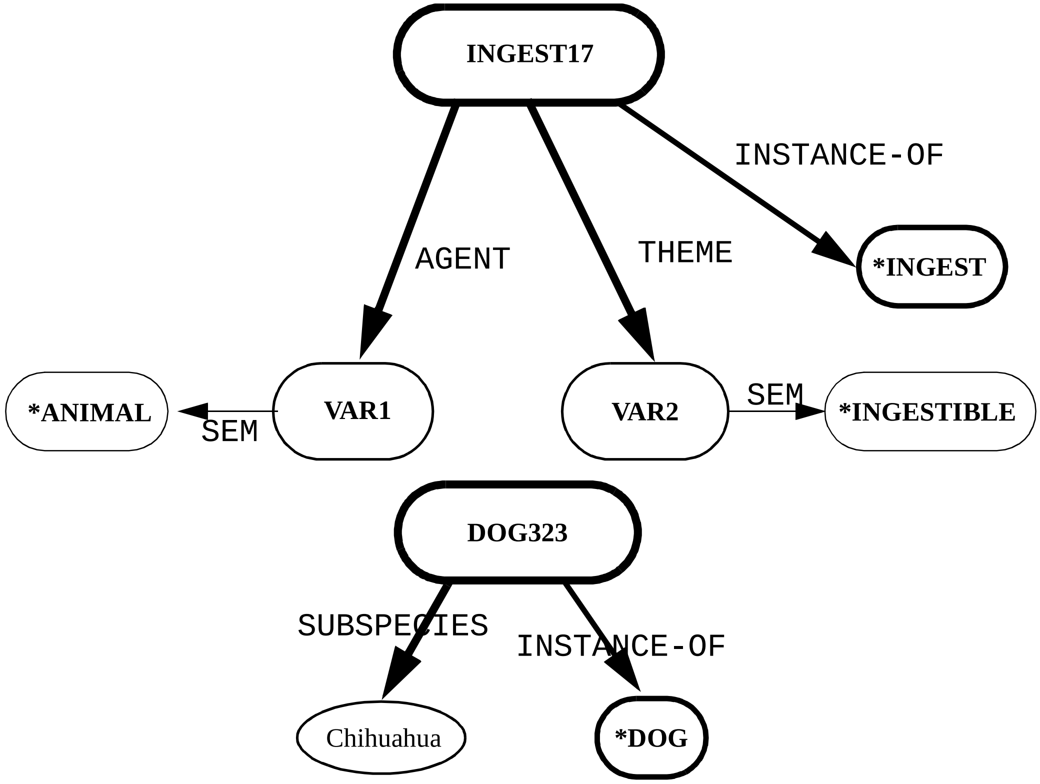


Figure 10A. Illustration of slot constraints and filler types (repeated from Figure 5B for convenience)

AGENT slot might be ANIMAL, and a candidate filler might be an instantiation of DOG; thus the graph must be traversed to find the best path from DOG to ANIMAL (in this case a trivial hierarchical traversal). In the trivial case of verifying that the candidate filler is in a subtree headed by the constraint, the graph is treated as a tree (i.e., non-taxonomic links are ignored); the cost of an IS-A arc is set to be very low, and the cost of a SUBLASSES arc (the inverse of the IS-A arc), as well as all other links, is set to very high. Thus the constraint satisfaction test is treated trivially.

In many cases, however, the simple IS-A test will fail, because the base constraints are established for literal meaning, whereas the input contains a meaning shift. Thus, in metonymic or metaphoric text the IS-A constraints fail. Then the graph traversal is expanded to include other, appropriately weighted, arcs (relations) in the ontology. The sorts of relations that are used in treating the cases of metonymy and (some) metaphor are among those additional relations in the ontology (in fact, they are included in the ontology often with the express purpose of helping to treat metaphors and/or metonyms). For example, in *The White House said yesterday...* the AGENT for *say* is a metonym; since the constraint for AGENT on the appropriate word sense of *say* is HUMAN, *White House* does not satisfy the trivial hierarchical constraint. Thus the shortest path that the graph search finds includes an OCCUPANT arc (inherited by all concepts below RESIDENCE in the ontology); traversing this arc identifies the likely existence of a occupied-for-occupant (or institution-for-member) metonymy. The traversal of this arc has a greater cost than the traversal of vertical hierarchical arcs, thus it wouldn't be preferred unless there were no uni-directional vertical path available.

For any two concepts in the ontology, there will be many possible paths between them, however, typically with different weights. The processing paradigm espoused here postulates that the best path between the two concepts will identify the correct relationship between them (if the weighting mechanism is appropriate and the relative weights are appropriately assigned). The example in Figure 10B and Figure 10C illustrates how two paths over the ontology can have differ-

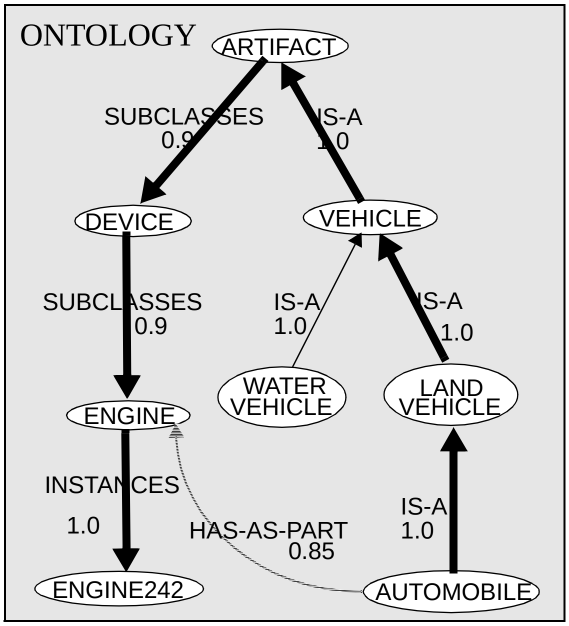


Figure 10B. Example of Ontology, with a path identified with bold arrow

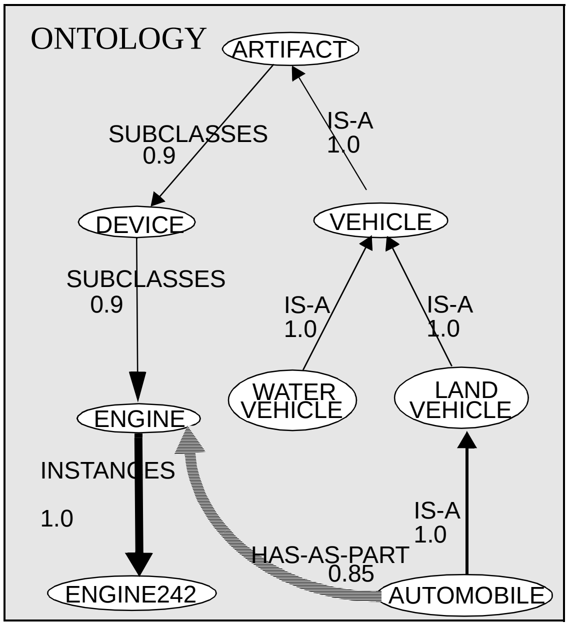


Figure 10C. Another view of the same ontology, but with a different path marked

ent weights, in this case, *.81* and *.85*, respectively. In the sentence *Fred drove his dual-cam V8 down Main street*, the phrase referring to the engine is used metonymically for the vehicle; in the semantic representation for *drive*, the constraint on what could be driven would specify the VEHICLE concept from the ontology. The two paths illustrated in these figures show how different weights on individual arcs lead to differing path weights (namely, *1.0 * 1.0 * 1.0 * 0.9 * 0.9 * 1.0 = 0.81* for the former, and *0.85 * 1.0 = 0.85* for the latter). If the arc weights are set appropriately, the shortest path from the filler to the constraint will reflect the metonymy, by traversing the arc capturing the part-for-whole relation embodied in the metonymic expression. It is clear from this example that the success of this approach is dependent on the richness of the ontology (not just in terms of concepts, but in terms of links as well) and on appropriate determination of weights.

11. Conclusion

The paradigm presented in this paper developed from the observation that the depth of analysis required for high-quality translation and other applications exceeds the capabilities of syntactic and shallow semantic levels. The information that is derived from the semantic analysis (which, in our terms, includes contextual semantics, pragmatics, stylistics, treatment of unexpected input, resolution of diectic phenomena, and other tasks, in addition to what has traditionally been called lexical semantics — that is, static, syntax-driven constraints on meaning) is represented in the language-neutral representation (the TMR). In practice, this depth of analysis requires substantial amounts of world knowledge for disambiguation and the other inferencing that is required in the process of building the TMR. The lexicon becomes the point in which much of this world knowledge is referenced and indexed. Therefore, the purpose of the **SEMANTIC-STRUCTURE** zone of the lexicon is to serve as the primary means of encoding lexical semantic units which are used to form the TMR; the ontology is used to define the semantics of the lexical semantic encoding, and, therefore, the semantics of the TMR as well. The purpose of all the other zones in the lexicon is, essentially, to assist in delivering these lexical semantic units; in other words, the purpose of the other zones is to provide static knowledge or to build dynamic knowledge sources which are used in the process of building TMRs.

12. Acknowledgements

Early work on this model of the lexicon also included members of the DIANA team and others: Lynn Carlson, Ingrid Meyer, Ralf Brown, Christine Defrise, Edward Gibson, Lori Levin. In addition, recent work on this model within the Mikrokosmos project also included Donalee Attardo, Salvatore Attardo, Steve Beale, Ralf Brown, Lynn Carlson, Betsy Cooper, JK Davis, Rod Johnson, Nicholas Ostler, Victor Raskin, Jerry Reno. *<<<should have DoD support etc>>>>*

13. Bibliography

Apresjan, Yu. (1974). “Regular Polysemy” in *Linguistics* vol. 142, pp. 5-32.

Apresjan, Yu. D., I. A. Mel’chuk, and A. K. Zholkovsky (1969). “Semantics and Lexicography: Towards a new type of Unilingual Dictionary”, in *Studies in Syntax and Semantics*, F. Kiefer, ed. Dordrecht: Reidel Publishing Company.

Bresnan, Joan, ed. (1982). The Mental Representation of Grammatical Relations. Cambridge MA:

MIT Press.

Brown, Ralf (1994). “FRAMEPAC”. Center for Machine Translation, Carnegie Mellon University.

Carbonell, J., T. Mitamura and E. Nyberg (1992). “The KANT Perspective: A Critique of Pure Transfer (and Pure Interlingua, Pure Statistics, ...)” in *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*. Montreal.

Carlson, Lynn and Sergei Nirenburg (1990). “World Modeling for NLP.” Center for Machine Translation, Carnegie Mellon University, Tech Report CMU-CMT-90-121.

Carlson, Lynn, Elizabeth Cooper, Ronald Dolan, Steven Maiorano (1994). “Representing Text Meaning for Multilingual Knowledge-Based Machine Translation” in *Proceedings of the First AMTA Conference*. Columbia MD.

Charniak, Eugene (1985). “A Single-Semantic-Process Theory of Parsing”. MS. Department of Computer Science, Brown University.

Cullingford, Richard and Boyan Onyshkevych (1987). “An Experiment in Lexicon-Driven Machine Translation” in *Machine Translation: Theoretical and Methodological Issues,* Sergei Nirenburg, ed. NY: Cambridge University Press.

Dorr, Bonnie (1993). *Machine Translation: A View from the Lexicon.* Cambridge MA: MIT Press.

Dorr, Bonnie, Joseph Garman, and Amy Weinberg (1994). “From Subcategorization Frames to Thematic Roles: Building Lexical Entries for Interlingual MT” in in *Proceedings of the First AMTA Conference*. Columbia MD.

Dorr, Bonnie and Clare Voss (1994). “ILustrate: a MT Developers’ Tool with a Two-Component View of the Interlingua” in *Proceedings of the First AMTA Conference*. Columbia MD.

Farwell, David, Louise Guthrie, and Yorick Wilks (1993). “Automatically Creating Lexical Entries for ULTRA, a Multilingual MT System” in *Machine Translation* vol. 8:3, pp. 127-146.

Fass, Dan (1988). “Collative Semantics: A Study in the Discrimination of Meaning.” Centre for Systems Science, Simon Fraser University. CSS/LCCR TR88-24.

Fass, Dan (1989). “Lexical Semantic Constraints.” Centre for Systems Science, Simon Fraser University. CSS/LCCR TR89-11.

Gibson, Edward (1990). “DIMORPH: A Morphological Analyzer.” Center for Machine Translation, Carnegie Mellon University. CMU-CMT-MEMO.

Gibson, Edward (1991a). “Bidirectional Active Chart Parsing.” Center for Machine Translation, Carnegie Mellon University. Draft.

Gibson, Edward (1991b). “BICHART: A Bidirectional Chart Parser.” Center for Machine Translation, Carnegie Mellon University. CMU-CMT-MEMO.

Hobbs, Jerry (1985). “Ontological Promiscuity,” in *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*. Chicago.

Hovy, Eduard (1988). *Generating Natural Language under Pragmatic Constraints*. Yale University, PhD dissertation.

Hovy, Eduard, Julia Lavid, Elisabeth Maier, Vibhu Mittal, Cecile Paris (1992). “Employing Knowledge Resources in a New Text Planner” in *Aspects of Automated NL Generation*, Dale, Hovy, Rosner, and Stock, eds. Lecture Notes in AI no. 587. Heidelberg: Springer

Verlag.

Hovy, Eduard and Elisabeth Maier (1994). “Parsimonious or Profligate: How Many and Which Discourse Structure Relations?” to appear in *Discourse Processes.*

Jackendoff, Ray (1983). *Semantics and Cognition*. Cambridge MA: MIT Press.

Jackendoff, Ray (1990). *Semantic Structures*. Cambridge MA: MIT Press.

Kamp, Hans (1981). “A Theory of Truth and Semantic Representation” in *Formal Methods in the Study of Language*, J. Groenedijk, J. Janssen, M. Stokhof, eds. Amsterdam: Mathematical Center Tracts.

Lakoff, George (1987). *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*. Chicago: University of Chicago Press.

Lakoff, George (1988). “Cognitive Semantics” in *Meaning and Mental Representations*, Umberto Eco, Marco Santambrogio, and Patrizia Violi, eds. Bloomington IN: Indiana University Press.

Lakoff, George and Mark Johnson (1980). *Metaphors We Live By.* Chicago: University of Chicago Press.

Levin, Beth (1989). “English Verbal Diathesis”. Lexicon Project Working Papers #32. Massachusetts Institute of Technology.

Levin, Beth (1991). “Building a Lexicon: The Contribution of Linguistics” in *International Journal of Lexicography.* vol. 4:3, pp. 205-226.

Levin, Lori and Sergei Nirenburg (1994). “Construction-Based MT Lexicons” in *Current Issues in Computational Linguistics: In Honour of Don Walker*, Antonio Zampolli, Nicoletta Calzolari, Martha Palmer, eds. Kluwer Academic and Giardini Editori E Stampatori in Pisa.

McDermott (1978). “Tarskian Semantics, or No notation without denotation!” in *Cognitive Science*. vol. 2:3.

Mel’chuk, Igor (1984). *Explanatory Combinatorial Dictionary of Modern Russian* (in Russian). Vienna: Wiener Slawistischer Almanach.

Meyer, Ingrid and James Steele (1990a). “The Presentation of an Entry and of a Super-Entry in an Explanatory Combinatorial Dictionary,” in *The Meaning-Text Theory of Language: Linguistics, Lexicography, and Practical Implications*, James Steele, ed. Ottawa: University of Ottawa Press.

Meyer, Ingrid, Boyan Onyshkevych, and Lynn Carlson (1990b). “Lexicographic Principles and Design for Knowledge-Based Machine Translation.” Center for Machine Translation, Carnegie Mellon University. CMU-CMT-90-118.

Mitamura, Teruko (1990). “The Hierarchical Organization of Predicate Frames for Interpretive Mapping in Natural Language Processing.” Ph.D. Dissertation. Center for Machine Translation, Carnegie Mellon University. CMU-CMT-90-117.

Monarch, Ira (1989). “ONTOS: Reference Manual.” Center for Machine Translation, Carnegie Mellon University. CMU-CMT-MEMO.

Nirenburg, Sergei, Jaime Carbonell, Masaru Tomita, and Kenneth Goodman (1992). *Machine Translation: A Knowledge-Based Approach.* San Mateo CA: Morgan Kaufmann Publishers.

Nirenburg, Sergei and Christine Defrise (1991). “Practical Computational Linguistics,” in *Computational Linguistics and Formal Semantics*, R. Johnson and M. Rosner, eds. Cambridge:

Cambridge University Press.

Nirenburg, Sergei and Kenneth Goodman (1990). “Treatment of Meaning in MT Systems,” *Proceedings of the Third International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language*. Linguistic Research Center, University of Texas at Austin.

Nirenburg, Sergei and Lori Levin (1992). “Syntax-Driven and Ontology-Driven Lexical Semantics” in *Lexical Semantics and Knowledge Representation*, James Pustejovsky, ed. Heidelberg: Springer Verlag.

Nirenburg, Sergei, Victor Raskin, and Allen Tucker (1987). “The Structure of Interlingua in TRANSLATOR” in *Machine Translation: Theoretical and Methodological Issues,* Sergei Nirenburg, ed. NY: Cambridge University Press.

Nyberg, Eric (1988). “The FRAMEKIT User’s Guide, Version 2.0.” Center for Machine Translation, Carnegie Mellon University. CMU-CMT-MEMO.

Nyberg, Eric and Teruko Mitamura (1992). “The KANT System: Fast, Accurate, High-Quality Translation in Practical Domains” in *Proceedings of COLING-92*. Trento.

Onyshkevych, Boyan and Sergei Nirenburg (1992). “Lexicon, Ontology, and Text Meaning” in *Lexical Semantics and Knowledge Representation*, James Pustejovsky, ed. Heidelberg: Springer Verlag.

Ostler, Nicholas and B.T.S. Atkins (1992). “Predictable Meaning Shift: Some Linguistic Properties of Lexical Implication Rules” in *Lexical Semantics and Knowledge Representation*, James Pustejovsky, ed. Heidelberg: Springer Verlag.

Pustejovsky, James (1991). “The Generative Lexicon” in *Computational Linguistics*, 17:4.

Schank, Roger (1973). “Identification of Conceptualizations Underlying Natural Language” in *Computer Models of Thought and Language*, Roger Schank and Kenneth Colby, eds. San Francisco: W.H. Freeman Co.

Skuce, Douglas and Ira Monarch (1990). “Ontological Issues in Knowledge Base Design: Some Problems and Suggestions.” Center for Machine Translation, Carnegie Mellon University. Technical Report CMU-CMT-119.

Small, Steve and Chuck Rieger (1982). “Parsing and Comprehending with Word Experts (A Theory and Its Realization),” in *Strategies for Natural Language Processing,* Wendy Lehnert and Martin Ringle, ed. Hillsdale NJ: Lawrence Earlbaum Associates.

Tulving, E. (1985). “How many memories are there?” in *American Psychologist.* vol. 40, pp. 385-398.

Wilks, Yorick (1973). “An Artificial Intelligence Approach to Machine Translation” in *Computer Models of Thought and Language*, Roger Schank and Kenneth Colby, eds. San Francisco: W.H. Freeman Co.

Wilks, Yorick (1975). “Preference Semantics” in *Formal Semantics of Natural Language*, E.L. Keenan, ed. Cambridge: Cambridge University Press.

Wilks, Yorick (1992). Review of Ray Jackendoff’s *Semantic Structures,* in Computational Linguistics, vol. 18:1 pp. 95-97.

Appendix A: Sample Lexicon Entries

The examples below illustrate some of the salient aspects of the lexicon entry structure. For brevity, these example entries are only partial, with various zones removed.

```
+eat-v1  :=
     CAT: V
     MORPH:
               STEM-V: (ate v+past)
                       (eaten v+past-part)
     ANNO:
               DEF: “ingest solid food through mouth”
     SYN:
               SYN-CLASS: (trans +)  ;redundant w/ syn-struc
     SYN-STRUC:
               SYN-S-LOCAL:  ;; this would actually be regular class member
                         ((root $var0)
                          (subj ((root $var1)
                                 (cat n)))
                          (obj  ((root $var2)
                                 (opt +)
                                 (cat n))))
     SEM-STRUC:
               LEX-MAP:
                         (%ingest
                            (agent (value ^$var1)
                                   (sem *animal))
                            (theme (value ^$var2)
                                   (sem *ingestible)
                                   (relax-to *physical-object)))

+kick-v7  :=      ;;”for the idiom “kick the bucket”
     CAT: V
     ANNO:
               DEF: “to die”
     SYN:
               IDIO-F: (idiomatic +)
     SYN-STRUC:
               SYN-S-LOCAL:
                         ((root $var0)
                          (subj ((root $var1)
                                 (cat n)))
                          (obj  ((root +bucket-n1)
                                 (null-sem +) ;contributes no semantics
                                 (cat n)
                                 (det ((root +the-det1)
                                       (null-sem +))))))
     SEM-STRUC:
               LEX-MAP:
                         (%die
                            (theme (value ^$var1)))
     PRAGM:
               STYL: (formality 0.1)   ;;extremely informal

+in-prep1  :=
     CAT: prep
     ANNO:
               DEF: “located within the confines of”
               EX: “the pen in the box”
               COMMENTS: “not the ‘destination’ sense”
     SYN-STRUC:
               SYN-S-LOCAL:
                         ((root $var1)   ;;what it attaches to
                           ;; syn. category not specified for what it attaches to
                          (pp-adjunct ((root $var0)
                                       (cat prep)
                                       (obj ((root $var2)
                                             (cat n))))))
     SEM-STRUC:
               LEX-MAP:
                         (^$var1
                            (instance-of
                               ;;can attach to events or objects
                               (sem (*OR* *object *event)))
                            (location
                               (value ^$var2)
                               (sem *physical-object)))

+abhor-v1  :=
     CAT: v
     ANNO:
               DEF: “hate very strongly”
               TIME-STAMP: 940820 victor
               LAST-MOD: 940902 boyan
     SYN:
               SYN-CLASS: (trans +)   ;redundant w/ syn-struc
     SYN-STRUC:  ;;this would actually be done by class membership
               SYN-S-LOCAL:
                   *OR*  ((root $var0)
                          (subj ((root $var1)
                                 (cat n)))
                          (obj  ((root $var2)
                                 (cat n)))) ;e.g., Pat abhors chaos
                         ((root $var0)
                          (subj ((root $var1)
                                 (cat n)))
                          (xcomp ((root $var2)
                                  (finite -)
                                  (cat v)))) ;e.g., Pat abhors drinking
     SEM-STRUC:
               LEX-MAP:
                         (%ATTITUDE
                            (type evaluative)
                            (attitude-value  < 0.1)
                            (scope ^$var2)
                            (attributed-to (value ^var1*)
                                           (default *speaker*)))
                         (^$var2    ;; constrain what can be abhorred
                            (instance-of
                               (sem *OR* *event *object)))
                         (^$var1    ;; constrain who can  abhor
                            (instance-of
                               (sem *animate)
                               (relaxable-to *object)))

     LEX-RULES:
               LR-LOCAL:
                  (LR#10 +abhorrent-adj1) ;;via -ent adj formation rule
                  (LR#6 +abhorrence-n1)
     PRAGM:
               STYL: ;;these values only need to be approximate
                         (force 0.7)
                         (simplicity 0.4)
                         (directness 0.6)

+abhorrent-adj1 :=
     CAT: adj
     ANNO:
               DEF: “strongly hated”
               COMMENT: “auto-generated from +abhor-v1 by LR#10”
               TIME-STAMP: 940820 victor
               LAST-MOD: 940902 boyan
               CROSS-REF: +abhor-v1
     SYN:
               IDIO-F: (attributive +)
                       (predicative +)
     SYN-STRUC:
               SYN-S-CLASS: adj-att-pred
     SEM-STRUC:
               LEX-MAP:
```

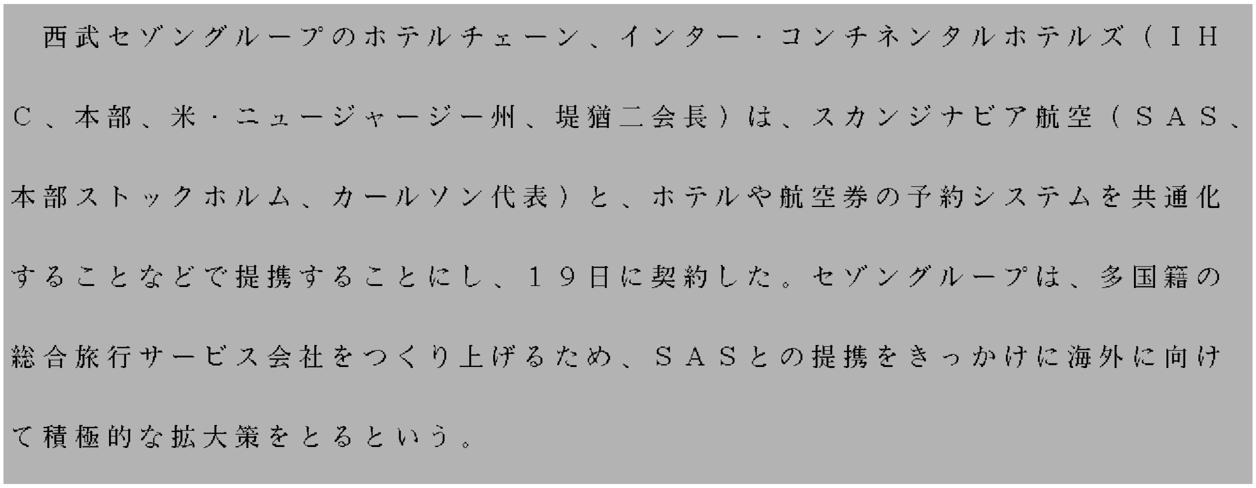
```
                         (%ATTITUDE
                            (type evaluative)
                            (attitude-value  < 0.1)
                            (scope ^$var1)
                            (attributed-to *speaker*))
     PRAGM:
               STYL: ;;these values only need to be approximate
                         (force 0.7)
                         (simplicity 0.4)
                         (directness 0.6)

+abhorrence-n1 :=
     CAT: n
     ANNO:
               DEF: “feeling of abhorring”
               COMMENT: “auto-generated from +abhor-v1 by LR#6”
               TIME-STAMP: 940820 victor
               LAST-MOD: 940902 boyan
               CROSS-REF: +abhor-v1
     SYN:
               IDIO-F:
                         (count -)
                         (proper -)
     SYN-STRUC:
               SYN-S-LOCAL:
                         ((root $var0)
                          (cat n)
                          (oblique1 ((root *OR +of-prep1 +for-prep1)
                                   ;; root doesn’t contribute semantics
                                     (null-sem +)
                                     (cat prep)
                                     (obj ((root $var1)
                                           (cat n)))))
                         ;; actually a disjunction to allow non-finite xcomp obj
     SEM-STRUC:
               LEX-MAP:
                         (%ATTITUDE
                            (type evaluative)
                            (attitude-value  < 0.1)
                            (scope ^$var1)
                            ;;default attrib-to can be overridden
                            ;;  by a possessive construction, for example
                            (attributed-to *speaker*))
                         (^$var1    ;; constrain what can be abhorred
                            (instance-of
                               (sem *OR* *event *object)))
```

```
LEX-REL:
          PAR-REL:
          SYN-REL:
PRAGM:
          STYL: ;;these values only need to be approximate
                    (force 0.7)
                    (simplicity 0.2)
                    (directness 0.6)
```

Appendix B: TMR Example

Below is a fragment of a TMR for an example text in Japanese from Asahi Shimbun (Morning Edition), 20 April 1989; a rough English gloss is presented below, and glosses are interspersed in the TMR for convenience. The TMR here is simplified for expository purposes (e.g., most reverse links are removed), and this TMR assumes a simplified ontology.



　西武セゾングループのホテルチェーン、インター・コンチネンタルホテルズ（ＩＨＣ、本部、米・ニュージャージー州、堤猶二会長）は、スカンジナビア航空（ＳＡＳ、本部ストックホルム、カールソン代表）と、ホテルや航空券の予約システムを共通化することなどで提携することにし、１９日に契約した。セゾングループは、多国籍の総合旅行サービス会社をつくり上げるため、ＳＡＳとの提携をきっかけに海外に向けて積極的な拡大策をとるという。

English Gloss: *The Seibu Sezon Group’s hotel chain, Intercontinental Hotels (IHC; headquartered in New Jersey in the United States; chairman, Tsutsumi Yuji), on the 19th contracted with Scandinavian Airlines (SAS; headquartered Stockholm; representative, Carlson), having decided to tie-up for such things as common (i.e. joint) use of a hotel and airline ticket reservation system. The Sezon Group, in order to set up a multinational comprehensive travel service company, will pursue an active overseas expansion policy by means of the tie-up with SAS.*

```
TMR ::=
     propositions:             %proposition_1 %proposition_2
                               %proposition_3 %proposition_4
                               %proposition_5 %proposition_6
     speech-acts:              %statement_1
     relations:                %coreference_1 %coreference_2
```

```
                              %coreference_3
                              %reason-domain-rel_1
                              %purpose-domain-rel_1
                              %purpose-domain-rel_2
                              %purpose-domain-rel_3
                              %result-domain-rel_1

%statement_1 ::=
     scope:                   %proposition_1 %proposition_2
                              %proposition_3 %proposition_4
     speaker:                 *author*
     hearer:                  *reader*

%time_0
     at                       890420    ;pubdate”
```

The Seibu Sezon Group’s hotel chain, Intercontinental Hotels (IHC; headquartered in New Jersey in the United States; chairman, Tsutsumi Yuji), on the 19th contracted with Scandinavian Airlines (SAS; headquartered Stockholm; representative, Carlson),...

```
%proposition_1 ::=
     head:                    %create_1
     time:                    %time_1
     aspect:                  %aspect_1

%create_1 ::=
     agent:                   %company_1
     theme:                   %contract_1
     accompanier:             %company_2

%company_1 ::=
     name:                    $”Intercontinental Hotels
     headquarters:            $“United States” (COUNTRY)
                              “New Jersey” (PROVINCE 1)
     alias:                   $“IHC”
     chairman:                %person_1
     owner:                   %company_3
     owner-of:                %set_1

%person_1 ::=
     name:                    $“Tsutsumi Yuji”

%contract_1

%set_1 ::=
     cardinality:             >1
     member-type:             *hotel
```

```
%company_2 ::=
     name:                    $“Scandinavian Airlines”
     headquarters:            $“Sweden” (COUNTRY)
                              “Stockholm” (CITY 1)
     alias:                   $“SAS”
     represented-by:          $“Carlson”  ;;shorthand for now

%company_3 ::=
     name:                    $“Seibu Sezon Group”

%time_1 ::=
     at:                      890419

%aspect_1 ::=
     phase:                   end
     iteration:               single
     duration:                momentary
     telic:                   true
```

". . . having decided to tie-up for such things as common (i.e. joint) use . . . ."

```
%proposition_2 ::=
     head:                    %decide_1
     time:                    %time_2
     aspect:                  %aspect_2

%decide_1 ::=
     agent:                   %company_1
     theme:                   %proposition_3

%aspect_2
     phase:                   end
     iteration:               single
     telic:                   true
     duration:                momentary

%reason-domain-relation_1 ::=
     arg1:                    %create-1
     arg2:                    %decide_1
```

;;the following represents that IHC created a tie-up

```
%proposition_3 ::=
     head:                    %create_2
     time:                    %time_3
     aspect:                  %aspect_3
```

```
%create_2 ::=
     agent:                   %company_1
     theme:                   %tie-up_1

%tie-up_1                     ;; a subtype of *agreement

%purpose-domain-relation_1 ::=
     arg1:                    %tie-up_1
     arg2:                    %set_2

%set_2 ::=
     cardinality:             >=1  ;;from “such things as”
     members:                 %utilize_1

%aspect_3 ::=
     phase:                   end
     iteration:               single
     telic:                   true
     duration:                momentary
```

;;” . . . common (i.e. joint) use of a hotel and airline ticket reservation system.”

```
%proposition_4 ::=
     head:                    %utilize_1
     time:                    %time_4
     aspect:                  %aspect_4

%utilize_1 ::=
     agent:                   %set_3
     theme:                   %reservation_system_1
     manner:                  $jointly

%reservation_system_1         ;;a shorthand
     reservation_type:        $hotel-stay $airplane_ticket

%set_3 ::=
     cardinality:             2
     members:                 %company_1 %company_2

%aspect_4 ::=
     phase:                   continue
     iteration:               multiple
     telic:                   true
     duration:                prolonged
```

The Sezon Group, in order to set up a multinational comprehensive travel service company, will pursue an active overseas expansion policy by means of the tie-up with SAS.

```
%proposition_5 ::=
     head:                    %implement_1
     time:                    %time_5
     aspect:                  %aspect_5

%implement_1 ::=
     agent:                   %company_4
     theme:                   %policy_1
     means:                   %tie-up_2

%purpose-domain-relation_2 ::=
     arg1:                    %create_3
     arg2:                    %implement_1

%company_4 ::=                ;; coref to company_3
     alias:                   $”Sezon Group”

%policy_1 ::= ;;shorthand for example
     policy-type:             $”active overseas expansion”

%tie-up_2 ::=
     owner:                   %company_5 ;coref to company_2

%aspect_5 ::=                 ;;telic unknown
     phase:                   continue
     duration:                prolonged

%company_5 ::=
     alias:                   $”SAS”
```

;;”. . . in order to set up a multinational comprehensive travel service company . . . .”

```
%proposition_6 ::=
     head:                    %create_3
     time:                    %time_6
     aspect:                  %aspect_6

%create_3 ::=
     agent:                   %company_4
     theme:                   %company_6

%company_6 ::= ;lots of shorthand here
     activity:                $” comprehensive travel”
     nationality:             $multi

%aspect_6 ::=
     phase:                   end
     iteration:               single
     telic:                   true
     duration:                momentary
```

```
%purpose-domain-relation_3 ::=
     arg1:                    %implement_1
     arg2:                    %create_3

%result-domain-relation_1 ::=
     arg_1:                   %decide_1
     arg_2:                   %create_1
```

;;The text was written (%time_0) after IHC and SAS contracted (%time_1).

```
%temp-rel_1 ::=
     type:                    after
     arg_1:                   %time_0
     arg_2:                   %time_1
```

;;They contracted (%time_1) after IHC decided to tie-up (%time_2).

```
%temp-rel_2 ::=
     type:                    after
     arg_1:                   %time_1
     arg_2:                   %time_2


%coreference_1: %tie_up_2 %tie-up_1

%coreference_2: %company_4 %company_3

%coreference_3: %company_5 %company_2
```

Appendix C: BNF for TMR

The notation below sketches the syntax of TMRs in a BNF-like notation. The notation `::=` is used to define the structure of frames; the notation `-->` identifies a rewrite rule or expansion.

```
<TMR>          ::=
     propositions:       <proposition> +
     speech-acts:        <speech-act> +
     stylistics:         <stylistic-factors> *
     relations:          (<text-relation> | <coreference>
                         | <temporal-relation>
                         | <quantifier-relation>
                         | <domain-relation> ) *

<proposition>  ::=
     head:               (<concept-instance> | <attitude> |
     <set>)
     aspect:             <aspect>
     time:               <time>*
     modality:           <modality>*
     attitude:           <attitude>*
```

```
<concept-instance>       ::=
     instance-of:        <<concept>>
               ; the frame is actually usually named by a gensym
               ; of the concept name
     [<<property-name>> :(<<concept>> | <concept-instance>
                          <<value>> | <set>)* ]*

               ;case roles and physical properties are among the most
               ;typical properties of concepts; all of those we
               ;expect to have been defined in the ontology

<<concept>>    -->       ONTOSUBTREE-OR(all)

               ;ONTOSUBTREE-OR is a function which returns a DISJUNCTIVE
               ;SET of all the elements in the ontological network
               ;rooted at its argument(s)
               ;Note that this function is not part of TMR, only of our
               ; description of it.


<<property-name>>-->     ONTOSUBTREE(property)

               ;ONTOSUBTREE returns a single terminal element of the
               ;subtree specified by its argument; in this case, returns
               ;tree of all possible properties


<aspect>       ::=
     aspect-scope:       <<scope>>
     phase:              begin | continue | end
     duration:           momentary | prolonged
     telic:              <<boolean>>
     iteration:          <numerical-value> | multiple

               ; the number of iterations can be explicitly stated (e.g.
               ; "twice") or just known to be multiple (e.g. "John hopped
               ; around on one foot").


<time>         ::=
     at:                 <<time-expression>>
     start:              <<time-expression>>
     end:                <<time-expression>>
     duration:           <<numerical-value>> {unit}

<<time-expression>> -->  (< | > | >= | <= | ) YYMMDD
```

```
<attitude>     ::=
     attitude-type:      <<attitude-type>>
     attitude-value:     [0.0, 1.0]
     attitude-scope:     <<scope>>
     attributed-to:      <<attributed-to>>
     attitude-time:      <time>

<attitude-type>::=       evaluative | saliency

               ;the number of attitude types may change

<<scope>>      -->       any TMR expression or set of such

<<attributed-to>> -->    ONTOSUBTREE-OR(intelligent-agent)

               ;any instance of the ontological type intelligent-agent

<modality> ::=
     modality-type:      <<modality-type>>
     modality-value:     [0.0,1.0]
     modality-scope:     <<scope>>

<<modality-type>>:-->    epistemic | deontic | volitive
                         | potential

<speech-act>   ::=
     speech-act-type:    <<speech-act-type>>
     speech-act-scope:   <<scope>> ;;usually a proposition
     speaker:            <<speaker-hearer>>
     hearer:             <<speaker-hearer>>
     time:               <time>

<<speech-act-type>>:-->  statement | question | ...

<<speaker-hearer>> -->   *speaker* | *hearer*
                         | ONTOSUBTREE-OR(intelligent-agent)

<stylistic-factors>::=
     [ <<style-factor>>: [0.0,1.0] ]*

<<style-factor>>-->      formality | politeness | respect
                         | force | simplicity | color |
                         directness

<<value>>      -->       <<numerical-value>> | <<literal-value>>

<<numerical-value>>      -->       any numerical expression

<<literal-value>>        -->       “string”
```

```
<set>          ::=
     member-type:        <<concept>> | <concept-instance>
     cardinality:        (< | > | >= | <= | ) <<numerical-expr>>
     elements:           <concept-instance> *
     complete:           <<boolean>>
     excluding:          <<concept>> | <concept-instance>
     subset-of           <set>
     multiple:           <<boolean>>
     indeterminate:      <<boolean>>
     proper:             <<boolean>>

<<boolean>>    -->       true | false

<coreference>  ::= <concept-instance> <concept-instance>+

<time-relation>::=
     type:               after | during
     arg1:               <time>
     arg2:               <time>
     value:              [0.0, 1.0]

<text-relation>::=
     type:               <<text-relation-type>>
     arg1:               <<text-relation-argument>>
     arg2:               <<text-relation-argument>>

               ;text relations are non-ontological relations which
               ; reflect relevant text structure (eventually to include
               ; discourse relations)


<<text-relation-type>>-->particular | reformulation
                         | progression | conclusion

<<text-relation-argument>> --> <proposition> +

<domain-relation>::=
     ;; actually, in implementation the type is usually prepended to the object name
     type:               <<domain-relation-type>>
     arg1:               <<domain-relation-argument>>
     arg2:               <<domain-relation-argument>>

<<domain-relation-argument>>--> (<concept-instance>
                         | <attitude> | <domain-relation> )*

<<domain-relation-type>> --> ONTOSUBTREE(domain-relation)
```