The Acquisition and Interpretation of Complex Nominals

Michael Johnston†, Branimir Boguraev§, James Pustejovsky†

† *Computer Science Department, Brandeis University, Waltham, MA 02254*

§ *Apple Computer, Inc., 1 Infinite Loop, Cupertino, CA 95014*

1 Introduction

Complex nominals present one of the greatest challenges in semantic analysis to date, particularly when viewed from the concerns of polymorphic expressiveness in natural language. From the perspective of the traditional lexicon designer, complex nominals are formed generatively and therefore do not merit explicit listing except when they are clearly non-compositional. In the spectrum of complex nominals, technical terminology occupies a special place, in that it is highly productive yet serves to encapsulate the essential concepts of a particular technical domain, therefore meriting inclusion in a technical glossary for that domain. Interpreting technical complex nominals (henceforth TCNs) — complex nominals with terminological status — generatively is not generally feasible because they are often composed of technical words for which conventional dictionaries will not provide coverage. As lexicons, and dictionaries, evolve from being static sources of word definitions to dynamic knowledge bases which function as resources for natural language technologies (such as NL-based information access and retrieval, machine translation, and natural language understanding), it will be increasingly important for them to support processing of technical documentation. Hence, they must serve not just the function of a lexicon but also that of a technical glossary. Given the frequency with which they are coined it is impractical to rely on manual listing of TCNs, and so it becomes critical that on-line lexical resources be able to acquire and interpret TCNs dynamically as text is processed.

Previous work on the interpretation of complex nominals is not sufficiently general to be applied to large-scale domain independent tasks such as this. Approaches taken in the artificial intelligence and computational linguistics literature (e.g. McDonald 1982, Alshawi 1987) crucially rely on the existence of domain models. Domain models can be successfully hand-coded for the task of complex nominal interpretation in limited individual domains. However, since our task is to provide domain-independent mechanisms for the lexical acquisition and interpretation of TCNs, we cannot presuppose the existence of a hand-crafted model for each domain we encounter. More recently, there has been renewed interest in TCNs; however, this interest to date focuses solely on the acquisition aspect of the problem (see, for instance, Chen and Chen 1994, Su *et al.* 1994, Justeson and Katz 1995).

Approaches taken to the interpretation of complex nominals in the linguistic literature (e.g. Levi 1978, Warren 1978), succeed in enumerating the different subtypes of complex nominals that show up in compounds but do not provide discovery procedures for disambiguating between these subtypes. Furthermore, these approaches presuppose the existence of a fully-articulated semantics for each of the words in a complex nominal, a situation that is not readily available for technical vocabulary.

Our work focuses on the development and application of techniques which enable determination of the interpretation of complex nominals in technical documentation, without presupposing a domain knowledge base or a predefined semantics for the elements of the complex nominal. This work is situated within the framework of the Generative Lexicon (GL) (Pustejovsky 1991, 1994). This paper describes our methodology for the acquisition of complex nominals from corpora, and how assuming a particular size and type of text (the *closed corpus* assumption), these acquired complex nominals and their subparts can be automatically assigned representations which capture aspects of their semantic interpretation.

We begin in Section 2 by introducing the distinction between *open* and *closed* corpora and the *closed corpus* assumption which facilitates determination of the interpretation. The aspects of the interpretation of a TCN which we intend to capture are introduced and exemplified in Section 3. Section 4 describes the linguistic processing that is employed in order to acquire TCNs from corpora. In Section 5, we describe the determination of the *local ontologies* which the TCNs within a document populate. In Section 6, the determination of GL entries for the TCNs within a closed corpus is discussed and some of our initial results are presented. Section 7 summarizes and concludes the paper.

2 Closed and Open Corpora

For the purpose of concept learning in fixed domains, it is conventional to assume some version of the "closed world hypothesis". Under this assumption, for every assertion which is not explicitly given in the knowledge base, its negation is assumed to be true for purposes of completeness of the model. Hence, for closed worlds such as in limited expert system applications or finite domains of interpretation, closure on literals that are not axioms in the domain is a valid method for completing the theory. For inferences made in open domains, however, closure by this method is unreliable and risky at best. Furthermore, for purposes of concept learning, feature induction without assuming a closed world model is intractable.

From the perspective of our current work, where the goal is to acquire semantic information associated with particular lexical items in text corpora, this means that induction must be limited to literal contexts within which a word occurs, namely, the positive instances. This is of course the standard assumption and procedure in the field of corpus-based lexicon acquisition, and it consequently restricts the types of induction techniques that are applicable to the corpus. We will argue, however, that this restriction need not apply uniformly across all corpus types. We will distinguish two classes of corpora, *open* and *closed*. This distinction refers to the nature of how a corpus is (a) structured, (b) acquired, and (c) maintained.

With respect to the first parameter, a corpus can be: a simple collection of (minimally) tagged structures (e.g. sentences), such as the Brown Corpus; a collection of tagged structures within larger tagged units (e.g. paragraphs and articles) such as the Wall Street Journal; or a set of hierarchical and cross-indexed objects such as user manuals. Concerning how a corpus is acquired, there may be a minimal condition constraining inclusion, such as "any article in the Wall Street Journal from 1989", up to the corpus being a well-defined artifact which accomplishes a very specific goal in conveying a body of information. This defines a corpus such as a dictionary or a user manual, which is the text it is by intentional design. Finally, with respect to the last parameter, corpora vary in how they are maintained over time. For the Wall Street Journal, AP Newswire, and Financial Times corpora, the corpus simply grows, with minimal checks on consistency of the form of the tagged text; for a user manual, any information added to the corpus would hopefully be consistent with information already provided to the reader in the earlier corpus.

Given this discussion, we classify a corpus such as the Wall Street Journal, as used in the ARPA-funded MUC and TIPSTER efforts, as an *open* corpus. The documents used in our experiments, according to the same considerations, will be classified as *closed* corpora. These are technical instructional documents written for a particular knowledge level of user, with a very specific depth of description of the information included. Because of the "intentionally closed" nature of these corpora, we are not restricted to positive data, regarding the semantic domain of interpretation for the relations and properties acquired through inductive methods. That is, the absence of a particular pattern with a specific lexical item can be interpreted as a negative instance for that word.

The theory for a *closed corpus*, therefore, will be complete if either every ground atom in the language or its negation is included. Hence, in the absence of a literal ϕ, in which a predicate P is predicated of a, we augment our database, DB, with all the negative ground literals (i.e., $\neg P(a)$) as though we believed those as strongly as we do the elements in the base set of beliefs. The *closed corpus assumption* (CCA) can be loosely defined as follows.

Let DB be the axioms of our theory. Following Reiter (1980), we can view $\mathcal{T}[DB]$ as the closure of DB under logical entailment. The CCA then would augment this with a set of presupposed beliefs, DB_{psup}. The *strong* CCA would be stated as follows:

(a) $\phi \in \mathcal{T}[DB] \Leftrightarrow DB \models \phi$.
(b) $\neg P \in DB_{psup} \Leftrightarrow P \notin \mathcal{T}[DB]$.
(c) $\phi \in CCA[DB] \Leftrightarrow \{DB \cup DB_{psup}\} \models \phi$.

Even with an intentionally closed corpus, however, this is too strong a statement on the assertability of "absent information" regarding objects in the domain; that is, any literal not predicating of a domain object would be taken as reason to conclude that the negation of that predicate holds for that object. As an example of this reasoning, with the sentence

All control panels are stored on the hard disk

and the absence of any mention of the predicate λx[`stored-on(x,hard-disk)`] for *application programs*, we would be forced to conclude, under the strong CCA, that $\neg$`stored-on(application,hard-disk)` is true. This is obviously false. For this reason, we will consider a family of weaker interpretations of the CCA, one of which we will assume for our subsequent discussion. Assume that there is a family of axioms in DB, $\mathcal{L}$, where closure is possible, while not being a property of the entire axiom set DB. The model might be stated as follows:

(a) $\phi \in \mathcal{T}[\mathcal{L}], \mathcal{L} \subseteq DB \Leftrightarrow \mathcal{L} \models \phi$.
(b) $\neg P \in \mathcal{L}_{psup} \Leftrightarrow P \notin \mathcal{T}[\mathcal{L}]$.
(c) $\phi \in CCA[\mathcal{L}] \Leftrightarrow \{\mathcal{L} \cup \mathcal{L}_{psup}\} \models \phi$.

Working with this assumption, then, the problem is now to accurately identify the subsets $\mathcal{L}_i$ under which closure is a confident operation.[1] What this says is that there is a set of axioms for which we can inference from closure, but that the operation is restricted to a structured subdomain, such as the sub-types of a type as defined in the knowledge base. This, in fact, corresponds to the case we are studying, where a noun and the compounds formed by the noun define a structured subset of axioms for purposes of inference under a closed corpus assumption.

3 The Analysis of TCNs

The target analysis of the TCNs within a document which this methodology can be utilized to capture encompasses the following four aspects:

1. The internal structures of the TCNs.
2. The semantic operations through which each TCN is composed.
3. The local ontologies into which related TCNs group.
4. The meaning of the TCNs and their component parts.
5. Cross-classificational structure licensed by relations among TCNs.

The internal structure of a TCN defines the order in which the subparts of the TCN compose with each other. This is represented by bracketing the component parts of the TCN into constituents. For example, a form N1 N2 N3 may be bracketed as [[N1 N2] N3] or [N1 [N2 N3]]. The form *sound control panel* could be bracketed [[*sound control*] *panel*] or [*sound* [*control panel*]].

There are a number of different semantic operations which may be involved when two constituents of a TCN compose. For example, in a form like *hard disk* the first part of

[1]This is explored in more detail in Pustejovsky and Johnston (in preparation).

the TCN *hard* acts as a sub-typing modifier, and specifies one of the formal properties of the disk. In a form like *start-up disk*, the initial part of the compound specifies the purpose to which the disk is put. The nature of the operations in a given TCN can be identified or at least narrowed down by inspection of their context in a *closed corpus*. Note that all the inferences associated with such inspection, examples of which follow, crucially depend on the validity of the weak version of the CCA assumed in the previous section. Local ontologies are hierarchies which capture subdomains of related objects within the content of a particular corpus. They group together TCNs which refer to objects of closely related types. For example, in the Macintosh Reference domain there is a local ontology of disks. The most basic type in this ontology is *disk*. The subtypes of *disk* are *floppy disk*, *RAM disk*, and *hard disk*. The type *hard disk* is further subtyped into *internal hard disk* and *external hard disk*. The identification of local ontologies provides critical information for the determination of inheritance relations between lexical entries.

In this work, in keeping with the tenets of Generative Lexicon theory, we view the meaning of a TCN and its component parts as being defined extensionally by the relations and properties it is composable with. Those relations and properties partially instantiate the qualia structure of the TCN. Also, the approach taken in GL to complex nominal interpretation is, as expected, one that enhances and amplifies the overall role played by each element in the complex nominal. For each local ontology, we derive partially instantiated GL entries for all of the TCNs and their subparts, such that through basic semantic operations of composition the more specific types can be derived from the more basic types.

As an additional organisational principle, providing an extra dimension to TCN interpretation, cross-classificational structure licensed by relations among TCNs reflects larger patterns of similarity or associativity of TCN use, exhibited across the entire text. In effect, this models the notion that semantically related TCNs are found in similar contexts, as well as that a family of TCNs closely related by function in the domain will be classified in a tight category. Thus, while methods for elaborating the local ontology for *disk* identify, among others, *RAM disk*, clustering techniques similar to those described in Waterman (1995), but applied to phrasally tagged syntactic contexts, place *RAM disk* in a category together with *battery power*, *powerbook computer*, *temporary storage device*, and *minimal system folder*.

In order to derive the resources on which TCN analysis is based the closed corpus in question undergoes a cascade of linguistic processing and syntactically and semantically driven knowledge mining. This processes involved are summarized in the next section. In Section 5, we provide examples of the ways in which the internal structure of TCNs is identified and the use of this and other information in deriving local ontologies.

4 Acquisition from Corpora

TCNs are identified in the text through the application of a suitably tuned grammar for technical terminology. The grammar is not dissimilar to the one presented in Justeson and Katz (1995); however, it should be stressed that a TCN, in the sense introduced earlier and discussed in this paper, subsumes the strong notion of a *technical term* as defined by Justeson and Katz — the primary difference is that for the purposes of acquisition and interpretation of TCNs it is essential to identify <u>all</u> nominal objects which map to entities in the domain. In order to pick out the relations and properties which apply to each TCN, the appearance of TCNs in a variety of basic syntactic patterns in the text is exploited. In order to identify these patterns within the text, the system needs a syntactic base which is more expressive than that provided by part-of-speech tagging alone. However, as it turns out, "shallow" syntactic parsing (as carried out for instance by the constraint grammar of Voutilainen *et al.*, 1992) offers enough in its system of syntactic function labels to drive the analysis of TCN context. The initial stage of the process is to run the shallow syntactic parser over the text. The next stage is to run a window over the text and identify matches between the text and a series of patterns which are tuned to identify higher syntactic constituents in which TCNs appear. Each of these constituents contributes a relation or property with which a TCN composes. The resulting stream of observations undergoes a series of further processes. Tagged text is removed and (largely) base forms are left in their place. Certain kinds of sortal anaphora are resolved. For example, uses of *disk* to refer to *hard disk* or *floppy disk* are identified and suitably expanded. The stream of observations is then collated into a catalog of TCNs. Associated with each TCN are lists of the relations and properties with which that TCN appears. The contents of this catalog are then assessed in order to derive the aspects of the analysis derived above.

5 Internal Structures and Local Ontologies of TCNs

In order to determine the internal structure of a given TCN, it is necessary to determine whether its subparts are themselves TCNs. For example, for the TCN *key chain access code*, the fact that the TCNs *access code* and *key chain* also appear in the corpus allows us to hypothesize that the internal structure is [[*key chain*][*access code*]][2]. This hypothesis can be further supported by examination of the relation and property sets associated with *key chain access code* and *access code*. If those for *key chain access code* include the majority of those for *access code*, then this is an appropriate structure for this TCN.

Comparison of relation and property sets across related TCNs also enables identification of the semantic opera-

[2] *key chain*, itself a valid contraction of *PowerTalk key chain*, refers to the ability, within a system-wide collaborative environment, of accessing an open set of diverse collaboration servives with a single log-on *access code*.

tions through which the TCN is composed. For example, if relations R1, R2, and R3 are found with [N1], and R1, R2, R3, R4, and R5 are found with [N2 N1], then we can hypothesize that [N2 N1] is a more specific kind of [N1]. Thus, since *floppy disk* shares relations with *disk* we conclude that *floppy disk* is a more specific kind of disk.

Once the internal structure of the TCNs has been identified, related TCNs can be grouped together and assembed into local ontologies. The basic structure of the local ontology is derived from similarities in form between the TCNs of which it is composed. For example, since *floppy disk, hard disk, RAM disk* all share the common head *disk* they are assembled into a local ontology of disks. Local ontologies are further verified, developed, and refined by inspection of the relations and property sets which are associated with each element. For example, if we establish from the corpus that *RAM disk*, *floppy disk*, and *hard disk* all share a series of properties and relations, we can confidently conclude that they are three subtypes of the kind *disk*. Identification of these local ontologies enables determination of some of the inheritance relations that hold between TCNs and will partially instantiate lexical inheritance hierarchies for the domain in question.

We turn now to the determination of lexical entries for the subparts of these TCNs which enable the assignment of interpretations to the TCNs within a given corpus.

6 Partially Instantiating Generative Lexica for TCNs

Having shown how local ontologies can be derived from the corpus, we turn now to the determination of the semantics of a TCN and its component parts. From the corpus analysis described in Section 4, we have an index of the TCNs attested and a record of the different relations and properties with which they appear. On the basis of the patterns which relations and properties form across local ontologies, semantic representations are assigned to the subparts of TCNs. In order to establish the basic semantic type of a given head, the lexical conceptual paradigms (henceforth LCPs) in which it appears are examined. These lexical conceptual paradigms are syntactic configurations and alternations which are prototypical of particular semantic types of entities (see Pustejovsky *et al.*, 1993). Once the basic semantic type of the head is established, its qualia structure and the qualia structures of the TCNs in which it appears are determined incrementally from the corpus.

One of the local ontologies established in our initial investigation of this methodology, using the Macintosh Reference corpus, is the disk ontology. We will present here a small portion of this ontology, which includes the TCNs: *floppy disk* and *hard disk*. The relation sets established from the corpus for these forms are as follows.

(1) *floppy disk* : `name [], erase [], access [], initialize [], use [], test [], save file on [], repair [], eject [], insert [] into disk drive, lock [], unlock [], protect []`

(2) *hard disk* : `name [], save file on [], repair [], erase [], access [], initialize [], use [], test [], attach [] to printer, reinstall system software on [], copy program disk to []`

These relation sets demonstrate the fact that, given that a corpus is closed, a strikingly accurate characterization of the domain can be established from corpus. In this initial investigation the system successfully established that floppy disks are things which can be inserted into a disk drive, ejected, locked, unlocked, and protected, while hard disks cannot, and that hard disks are things which can be attached to printers, have system software reinstalled on them, and have program disks copied to them, while floppy disks are not. Given this information, the interpretation of *floppy disk* and *hard disk* proceeds as follows.

From investigation of the attested TCNs, *disk* is established as a head noun. The basic lexical semantics for *disk* is established as follows. The TCNs *floppy disk* and *hard disk* appear as the object of prepositions such as *to*, *from*, and *on* in the corpus. The `container` LCP relates appearance in these syntactic positions to the qualia structure assigned to containers. TCNs headed by *disk* are also found in configurations such as:

(a) *store information on* [*disk*]
(b) *destroy* [*the information on a disk*]

From this, the system can infer that disk is in fact a specific subtype of container, the `information-phys-object-container`. This type is assumed to be a member of an inventory of general types which form the core of a generative lexicon. The noun *book* is of this type. The basic lexical semantics assigned to *disk* is as follows:

$$\left[\begin{array}{ll}
\textbf{disk(x,y)} & \\
\text{ARGSTR} = & \left[\begin{array}{l} \text{ARG1} = \texttt{x:information} \\ \text{ARG2} = \texttt{y:phys\_obj} \end{array}\right] \\
\text{QUALIA} = & \left[\begin{array}{l} \textbf{information-phys\_obj-container-lcp} \\ \text{FORM} = \texttt{hold(y,x)} \end{array}\right]
\end{array}\right]$$

Having established the basic semantic type of *disk* the next stage is to instantiate the qualia structure of *disk*. In order to identify the set of relations which are appropriate for *disk* we take the intersection of the relations sets for the technical complex nominals which are headed by *disk*.[3] This gives us the set: `access [], erase [], name`

[3]The *relation sets* for a lexical item, w_i, are derived from our statistical analysis of how TCNs behave in the closed corpora discussed in Section 2. Let N_i be assigned type τ. For each $[N^* N_i]$ compound satisfying bracketing conditions, assign it to a subtype $\tau' \leq \tau$. The set of axioms formed by τ and its subtypes $\{\tau_i'\}$ is identified as one of the well structured subsets $\mathcal{L}$ mentioned in Section 2. For $\mathcal{L}$ we wish to:

i. Identify the core features for the lexical item w_i.

`[],initialize [],use [],test [],save file on [], repair []`. This set defines what can happen to all disks and to a large extent defines extensionally the meaning of the term *disk* in this domain. We will call this set of relations the *core qualia* (CQ) of a lexical item. The core qualia are the relations which are found with all of the members of a local ontology. The partially instantiated semantic representation which we assign to *disk* is as follows:

```
disk(x,y)
ARGSTR = [ ARG1 = x:information
           ARG2 = y:phys_obj ]
QUALIA = [ information-phys_obj-container-lcp
           FORM = hold(y,x)
           CQ = {access [], erase [],
                 name [], initialize [],
                 use [], test [],
                 save file on [],
                 repair []} ]
```

Further analysis and inference is necessary in order to classify the different relations under different qualia, to identify the arity of the relations involved, and to link up the arguments of the relations with the argument structure of *disk*. Solutions to these aspects of the derivation of the representation are being actively investigated and will feature in future work. In addition to core qualia, we also utilize a category of *discriminant qualia* (DQ). These are the relations which are unique to a TCN which are not shared with other members of the local ontology. For example, the discriminant qualia for *floppy disk* are the set: `eject [],protect [],insert [] into disk drive,lock [], unlock []`. The lexical entry for *floppy disk* is as follows:

```
floppy_disk(x,y)
ARGSTR = [ ARG1 = x:information
           ARG2 = y:phys_obj ]
QUALIA = [ container-lcp
           FORM = hold(y,x)
           CQ = {access [], erase [],
                 name [], initialize [],
                 use [], test [],
                 save file on [],
                 repair [] }
           DQ = {eject [], protect [],
                 insert [] into
                 disk drive,
                 lock [], unlock [], } ]
```

Similarly the discrimant qualia for *hard disk* are: `attach [] to printer, reinstall system software on [],copy program disk to []`, and the resulting lexical entry is as follows:

ii. Semantically discriminate among these features; i.e., distinguish the features as values of particular qualia.

```
hard_disk(x,y)
ARGSTR = [ ARG1 = x:information
           ARG2 = y:phys_obj ]
QUALIA = [ container-lcp
           FORM = hold(y,x)
           CQ = {access [], erase [],
                 name [], initialize [],
                 use [], test [],
                 save file on [],
                 repair [] }
           DQ = {attach [] to printer,
                 reinstall system
                 software on [],
                 copy program disk
                 to []} ]
```

In order to determine a semantics for *floppy* and *hard* we compare the relation sets for *floppy disk* and *hard disk* with their intersection and their union. Given the *closed corpus assumption*, if a relation, R, for a TCN, T, does not occur in its relation set, then we can with confidence treat the absence of R with T as a "negative instance" of R for T. Therefore, since *hard disk* lacks *lock, unlock, insert [] into disk drive, eject* , and *protect* we can infer that part of the semantics of *hard* in this configuration is that it is not compatible with these relations. Since *hard disk* does have *attach [] to printer, reinstall system software on []*, and *copy program disk to []* we infer that part of the semantics of *hard* is that it adds these relations. Similarly, the semantics for *floppy* are that the head it modifies cannot be attached to a printer, have a program disk copied to it, or have system software reinstalled on it, but can be locked, unlocked, inserted into a disk drive, ejected, or protected.

On the basis of these differences between members of the local ontology, *hard* and *floppy* are assigned the following representations as modifier LCPs.

```
floppy(x)
ARGSTR = [ ARG1 = x:disk ]
QUALIA = [ modifier-lcp
           DQ = {eject [], protect [],
                 insert [] into
                 disk drive,
                 lock [], unlock [], } ]
```

```
hard(x)
ARGSTR = [ ARG1 = x:disk ]
QUALIA = [ modifier-lcp
           DQ = {attach [] to printer,
                 reinstall system
                 software on [],
                 copy program disk
                 to []} ]
```

These forms compose with the entry for *disk* to provide interpretations for *floppy disk* and *hard disk*.

7 Conclusion

The approach outlined here instantiates — for a specific lexical type, and the on the basis of a specific corpus type — the model of corpus-driven lexical semantics acquisition presented in Pustejovsky *et al.*, 1993. While the comprehensive analysis of corpora required by that framework is still outside the capabilities of current NLP technology, the notion of targeted acquisition driven by an appropriately selected corpus is applicable to a range of lexical types other than compounds, all of which require richly instantiated lexical entries. Furthermore, we found that the derived semantics for interpretation of a class of complex nominals facilitates other aspects of processing, including the resolution of anaphora and the interpretation of possessives, post-nominal complements, and conjoined nominals. This work suggests that a dynamic approach to lexicon, and dictionary, design need not be a theoretical position only, and that given the appropriate processing and analysis of suitable corpora, it is in fact a realizable goal.

References

Alshawi, Hiyan. 1987. *Memory and Context for Language Interpretation*. Studies in Natural Language Processing. Cambridge University Press, Cambridge, England.

Chen, Kuang-Hua, and Hsin-Hsi Chen, 1994. "Extracting Noun Phrases from Large-Scale Texts: A Hybrid Approach and its Automatic Evaluation". *Proceedings of the 34th Annual Meeting of the ACL*. Las Cruces, New Mexico.

Justeson, John, and Slava Katz, 1995. "Technical Terminology: Some Linguistic Properties and an Algorithm for Identification in Text". *Natural Language Engineering*. 1.1.

Levi, Judith N. 1978. *The Syntax and Semantics of Complex Nominals*. Academic Press, New York.

McDonald, David B. 1982. *Understanding Noun Compounds*. CMU Technical Report CS-82-102.

Pustejovsky, James. 1991. "The Generative Lexicon". *Computational Linguistics*. 17.4.

Pustejovsky, James. 1994. Linguistic Constraints on Type Coercion, in P. Saint-Dizier and E. Viegas (Eds.), *Computational Lexical Semantics*, Studies in Natural Language Processing. Cambridge University Press, Cambridge, England.

Pustejovsky, James, Sabine Bergler, and Peter Anick. 1993. Lexical Semantics Techniques for Corpus Analysis. *Computational Linguistics*. 19.2.

Pustejovsky, James, and Michael Johnston. (in preparation) *On Valid Inferences from Closed Corpora*. ms., Brandeis University.

Reiter, R., 1980. "Equality and Domain Closure in First-Order Databases". *Journal of the Association for Computing Machinery*. 27:235-249.

Su, Key-Yih, Ming-Wen Wu, and Jing-Shin Chang, 1994. "A Corpus-Based Approach to Automatic Compound Extraction". *Proceedings of the 34th Annual Meeting of the ACL*. Las Cruces, New Mexico.

Voutilainen, Atro, Juha Heikkila and Arto Antilla. 1992. *Constraint Grammar of English: A Performance-Oriented Introduction*. University of Helsinki. Publication No. 21.

Warren, Beatrice, 1987. *Semantic Patterns of Noun-Noun Compounds*. Gothenburg Studies in English 41. Acta Universitatis Gothoburgensis, Gothenburg.

Waterman, Scott, 1995. "Distinguished Usage", in *Corpus Processing for Lexicon Acquisition*, B. Boguraev and J. Pustejovsky (Eds.), MIT Press, Cambridge (in press).