The Role of Lexical Knowledge in Biomedical Text Understanding

Alexa T. McGray, Jeffrey L. Sponsler, Brandon Brylawski, Allen C. Browne

Lister Hill National Center for Biomedical Communications
National Library of Medicine; Bethesda, Maryland 20894

This paper discusses the design and development of a natural language understanding system for the biomedical domain. In particular, we describe the role that lexical information plays in such a system. We describe aspects of our coding system for lexical entries and discuss our approach to handling some of the challenges presented by developing a lexicon for this domain.

Automated understanding of natural language is an active area of research in Artificial Intelligence. The goal of making computers capable of understanding natural languages has been a common concern of many computer scientists, linguists, psychologists and philosophers. In fact, the last decade has seen the growth of Cognitive Science, a new interdisciplinary field devoted primarily to research in cognition, information processing and language processing both by humans and computers. This research is motivated by theoretical goals on the one hand:

> "...viewing the generation, interpretation, and acquisition of language as a computational process provides new constraints for language research. It takes the *deus ex machina* out of linguistic theory and forces the investigator to explain in empirically verifiable detail *how* meaning is extracted from surface structure." [15]

and practical goals on the other:

> "If computers could understand what people mean when people type (or speak) English sentences, the systems would be easier to use and would fit more naturally into people's lives." [1]

Our work is motivated by a desire to address fundamental issues in language analysis with the ultimate practical goal of developing natural language systems for intelligent storage and retrieval of biomedical information. Our work is based on the assumption that systems combining domain knowledge with sophisticated linguistic analysis will lead to improved representation and retrieval of biomedical information. Such systems will need to include linguistic knowledge: lexical information, and rules of morphology, syntax, and semantics, as well as knowledge about the domain: facts about the domain, relations among the facts, and rules to process these facts and relations. Additionally, and crucially, the systems must have a set of heuristics for use when dealing with incomplete or conflicting information.

Our work to date has concentrated on designing the linguistic aspects of a natural language understanding system for the biomedical domain. We have been concerned, in particular, with the role of lexical information in such a system. We view the lexicon as an important knowledge base. It provides information about the words and phrases of the language. This information is used by the syntactic and semantic rules as they build structured representations of the sentences of the language. For example, it is only because we know the differences in the lexical properties of the two verbs *promise* and *remind* that we can understand the differences in the interpretation of these two sentences: *They promised the students not to reveal their test results to anyone;* and *They reminded the students not to reveal their test results to anyone.* In the first sentence the implied, or underlying, subject of the verb *reveal* is *they.* In the second sentence the underlying subject of *reveal* is *the students.* The sentences have the same **surface** structure: *subject + verb + object + adverb + infinitive phrase,* but the interpretation or **underlying** structure is different. The lexical entries for these two verbs capture the difference: *promise* is marked as a *subject control* verb, and *remind* is marked as an *object control* verb.

The lexicon required for our domain is necessarily a specialized one. A specialized lexicon for parsing is required whenever the language to be parsed reflects a particular, specialized subject area. This can be as specific as *the language of weather bulletins,* or *the language of aviation hydraulics manuals,* or as broad as *the language of physics* [8]. It is generally noted that these specialized languages, or *sublanguages,* as they are sometimes called, differ from general English in their syntax and in their lexicons. A common view is articulated by Grishman & Kittredge [6]: "The variety of language used in a given science or technology not only is much smaller than the whole language, but is also more clearly systematic in structure and meaning." (See also Sager [18,19] for a similar view). This adequately characterizes a specialized language which is highly restricted in its syntax, word usage, and semantic constructs **and** which is used only in highly constrained text structures. However, when one considers specialized languages as broadly defined as *the language of physics,* or *the language of biomedicine,* then Lehrberger's [9] analysis appears to be more accurate: A *specialized* language *intersects* with the *standard* language, both of which are subsets of the *whole* language. Thus, natural language processing systems designed for these domains need to capture the regularities of general (standard) English syntax in addition to accounting for the peculiarities of the specialized language. Similarly, the lexicon will need to include general English lexical items as well as domain specific lexical items. In some cases, the distributional properties, as well as the multiplicity of senses for a general English word or term, can be reduced in the specialized lexicon. For example, it is unlikely that *organ* would refer to a musical instrument in the medical domain. The more restricted the domain is, the more reliable this becomes. (See, for example the discussion in Hobbs [7].)

The first step in the process of building our lexicon has been to take advantage of the rich source of information offered by MEDLINE. Each of MEDLINE's citation records includes a title, an abstract when available, author and journal names, and a set of Medical Subject Headings (MeSH). There are fifteen thousand MeSH headings and many more so-called **entry terms** which are either synonyms of the MeSH terms or closely related terms. We have written a number of word frequency and collocation programs to manipulate sets of abstracts grouped by subject area. The output of these programs is used for gathering syntactic, morphological, and word level information. (See Macdonald [10] for discussion of some interesting uses of the output of programs of this type.) The word frequency programs give both overall frequency information, and relative frequency across texts. This gives us some sense of the word's frequency of use over the whole subject area. Our collocation programs, with a threshhold of five adjacent words, suggest potential candidates for multi-word entries. The output of the collocation program is automatically checked against the file of MeSH terms and entry terms; matches are added to the output of the frequency programs. The remaining, non-MeSH phrases are reviewed manually to determine if they are potential candidates for multi-word entries in the lexicon.

The question of whether a phrase should be entered as a multi-word entry in the lexicon is complex. Phrases range from frozen Latin forms such as *lacertus fibrosus* to eponyms such as *Mendeleev's law* to others such as *pulmonary hypertension, mental spine, mitral valve prolapse,* and *benign central nervous system tumor*. The issue is whether to handle these phrases as single lexical items or as syntactic phrases whose meaning is derived from the separate words in the phrase. We have adopted the following approach. If the phrase is a frozen Latin form or an eponym, then it is considered to be a single lexical item. Further, if the phrase appears as a single concept in a standard, well-regarded medical dictionary *(Dorland's Illustrated Medical Dictionary* [5]), or if it is a MeSH term (with some exceptions), then the phrase is entered as a single multi-word lexical item; otherwise it is not. The terms *pulmonary hypertension* and *mental spine* are listed as single concepts in Dorland's, and *mitral valve prolapse* is a MeSH term. The phrase *benign central nervous system tumor* has a MeSH term embedded in it: *central nervous system.* This term is listed as a phrase in the lexicon and *benign* and *tumor* are listed as separate entries.

Each lexical entry encodes various types of syntactic and semantic information. Syntactic information includes information about syntactic category, inflectional variants, allowable complements, and allowable transformations. The semantic information currently includes information for logical interpretation. Further, if the term is a MeSH or MeSH entry term, it will be marked as such. This will facilitate the anticipated future use of the MeSH structure as a knowledge base for our system. Although we recognize the difficulties inherent in creating adequate knowledge bases for a domain as large as biomedicine (see Carbonell [3] for some discussion), we believe that the MeSH thesaurus can provide much of the required knowledge for our natural language understanding system. MeSH terms are organized in a structure consisting of fifteen top level nodes with up to seven levels of embedding. Top level nodes include *anatomical terms, organisms, diseases, chemicals and drugs, analytical, diagnostic and therapeutic technics and equipment, psychiatry and psychology, biological sciences,* and *physical sciences.* In addition to the MeSH terms themselves, there is a set of seventy-six subheadings which serve to qualify the terms. Examples are *etiology, adverse effects, therapeutic use, chemically induced, complications,* and *drug therapy.* An obvious limitation of the MeSH structure as it now exists is that the relationships between terms in the hierarchy are not explicitly marked. Many can be interpreted as *isa* links, e.g. ACETOBACTER *isa* GRAM-NEGATIVE AEROBIC BACTERIA, others as *partof* links, e.g. SEBACEOUS GLANDS *partof* SKIN, but many others are not as easily specified, e.g., APPETITE appears as a daughter of PSYCHOPHYSIOLOGY. In spite of these limitations, the MeSH thesaurus provides an extraordinary amount of domain knowledge. In the next phase of our work we intend to augment portions of the MeSH structure with relational links, thus allowing us to test the hypothesis that it can serve as a knowledge base for our system. In the interim, we are building a network of MeSH and MeSH entry terms in the lexicon. For example, the following terms are all entry terms for the MeSH term, NECROTIZING ULCERATIVE GINGIVITIS: *ulcerative stomatitis, trench mouth,* and *Vincent's infection.* Accordingly, each has as part of its lexical specification a logical link to the MeSH term.

We have developed a tool to ease the difficult and time-consuming task of building the lexicon. LEXTOOL is a menu-based program written in Quintus Prolog. It accepts as input the list of words and phrases that were created by the frequency analysis programs together with the manual review of the potential multi-word entries. With the interactive aid of the user, it gives as output a set of fully specified lexical entries. We have developed a coding system that is closely tied to that used by the *Longman Dictionary of Contemporary English* [17] (LDOCE). We have modified LDOCE's scheme somewhat, but we find that adherence to the basic coding system considerably speeds our building of the general English entries.

An example will illustrate the use of LEXTOOL. Assume that the word to be added to the lexicon is *induce.* The user is prompted to select a syntactic category, e.g., *noun, verb, adj,* etc. The user selects *verb.* Next the user is presented with a list of possibilities for type of inflectional variation: *reg* (regular), *regd* (regular, double the final consonant), or *var* (irregular variants). The choice will be *reg* since this verb has the regular variants *induce, induces, induced, inducing.* After specifying the variant information, the user is asked to provide information about the possible complement types of the verb. The set of possibilities is *intran* (intransitive), *link* (linking), *tran* (transitive), *ditran* (ditransitive), *cplxtran* (complex transitive). Depending on which of these codes is chosen, the system presents a list of possible complements (e.g., *np, adj, advbl, infcomp, ingcomp).* In this case, *tran* and *cplxtran* are chosen as complement types. The possible complement in the transitive case can be a noun phrase (np), e.g., *Too much food induces* **sleepiness.** The possible complements in the complex transitive case can be a noun phrase followed by an infinitive phrase (infcomp), e.g., *The risk* *of cancer induced* **him to give up smoking.** Next the user is presented with a list of codes indicating what the interpretation of the missing subject of the infinitive phrase should be. In this case, the code *objc* (object control) will be chosen, indicating that *him,* the object of the verb *induce,* is also the underlying subject of the lower verb, in this case, *give up.* Finally, the user is presented with a list of codes for lexically conditioned transformations. In this case, nothing is chosen. The verb *induce* may appear in the passive form, as in: *The patient was induced by a mixture of thiopental and curare,* but since this is the default case for transitive verbs, the verb is not specially marked. When all of the above steps have been taken, the system presents the finished entry to the user for verification. In this case it is: *induce(verb, reg, tran(np), cplxtran([np,[infcomp]), interp(objc)).* The user may then go on to process the next entry in the list.

The lexicon serves as the base for our parser. The parser is written in Quintus Prolog and runs on the Vax 11/780 under Unix, BSD 4.3. The input to the parser is a text and the output is a syntactic structure, or labeled parse tree. The trees represent surface constituency structure and include some semantic information. The semantic information is *projected* (in the sense of Chomsky [4]) from the lexicon; that is, the subcategorization properties of the lexical items are revealed in the syntactic structure. This means that a complement which appears out of its normal position will be linked to its underlying role. For example, in *Iron seems to play an indirect role in the decrease of this enzyme,* the surface subject of the higher clause is linked to its role as subject of the lower clause, since *seem* is a subject-raising verb. In *What were the children given by the physician's assistant?* the phrases *what, children,* and *physician's assistant* are linked, respectively, to their underlying positions as direct object, indirect object, and subject.

Lexical lookup occurs during the parsing process. Once a word has been looked up and found in the main lexicon, it is entered into the Prolog database as a lexical **frame,** thereby simplifying the work of the parser in accessing the relevant information in the entry. A frame is a data structure that is conceptually a set of triplets of the form [object, attribute, value]. We have designed and implemented a frame representation language in Quintus Prolog. Included in the language are procedures for creating and deleting frames, creating and deleting slots, storing and retrieving values from slots, and inheriting values from parent frame to child frame. Lexical frames consist of a variable set of attributes depending on the specification of the item in the main lexicon. Recall the lexical entry for the verb *induce: induce(verb, reg, tran(np), cplxtran([np, [infcomp]), interp(objc)).* The frame representation for this verb is as follows:

```
<<induce:
      instance: verb
      base: induce
      sing: induces
      past: induced
      past_part: induced
      pres_part: inducing
      tran: [np]
      cplxtran: [np,[infcomp]]
      interp: [objc] >>
```

Note that as part of the conversion procedure from a lexical entry to a lexical frame, morphological rules have generated the inflectional variants of the verb. For nouns, this means that the rules are sensitive to the *reg* and *glreg* (greco-latin) distinction. Compare, for example, *rebellion, rebellions* (reg) and *ganglion, ganglia* (glreg). Many medical words allow both greco-latin plurals and regular English plurals, e.g. *ganglia* or *ganglions.* If this is the case, the word is given both codes. Once an item has been looked up, the lexical frame is loaded into the Prolog database, and remains there during the current processing session. Then, if the same item, or an inflectional variant of the item is encountered during the same session, the main lexicon files do not need to be accessed again.

The grammar formalism for our parser is an extended Definite Clause Grammar (DCG). DCG's express the rules of the natural language as Horn clauses, which are a subset of predicate logic. There are several advantages to using a logic grammar formalism for natural language parsers (see Pereira [16], and McCord [11,12] for some discussion). One of the obvious advantages is that the rules of the language can be stated naturally and elegantly, serving both as a **description** of the language and as a description of the **procedure** for executing the program. Additionally, the power of unification (pattern matching) in Prolog, together with the power of the logical variable allows the expression of context sensitivity in a linguistically natural manner. For example, subject-verb agreement is context dependent. That is, the number of the verb **depends** on the number of its subject. A fragment from a simplified DCG will illustrate:

```
sentence(s(NP,VP),X,Z) :-
        np(NP,Num,X,Y),
        vp(VP,Num,Y,Z).

np(np(Noun),Num,X,Y) :-
        noun(Noun,Num,X,Y).

vp(vp(V,NP),Num,Y,Z) :-
        verb(V,Num,Y,W),
        np(NP,_,W,Z).
```

Terms beginning with uppercase letters are variables; those beginning with lower case are constants. The first rule expresses that a sentence is analyzed as a *np* (subject) and a *vp* (predicate). The first argument of each rule builds the structure for the sentence; i.e., a sentence structure is: s(structure of noun phrase, structure of verb phrase), while the noun phrase structure is simply: np (noun). The last two variables in each clause handle the string of words to which each rule must apply. For instance, the rule for sentence says that a sentence is a list going from point X to point Z if the list from point X to some point Y is a *np* and the list from point Y to point Z is a *vp.* Of interest here is the variable *Num.* Note that *Num* is included both in the subclause for the *np* subject as well as in the subclause for the *vp.* The fact that variables with the same name in the same Horn clause must have the same value ensures that subject-verb agreement is enforced. Take the simple example: *Food induces sleepiness.* The syntax rules given above would analyze this sentence as follows. First they would try to build a subject noun phrase. The lookup procedure would check to see that *food* is a noun. As part of the lookup procedure, all the information in the lexical entry is returned for use by the parser in building the sentence structure. Included is the information that *food* is singular, and, thus *Num* is given that value. Next, the rules try to build the vp predicate. The lookup procedure checks to see that *induce* is a verb **and** that its number is singular. This succeeds and the parser goes on to build the structure of the object noun phrase.

We have made a number of enhancements to the basic DCG formalism. For example, we have written a grammar rule translator which automatically builds syntactic structures without the need for explicit variables in the syntax rules. This simplifies the task for the writer of the grammar rules and allows these rules to express only linguistically relevant information. The rule fragment illustrated above is now simply as follows.

```
sentence ==>
   np(Num),
   vp(Num).

np(Num) ==>
   noun(Num).

vp(Num) ==>
   verb(Num),
   np(_).
```

The underscore in the second subclause of the vp rule indicates that an object need not agree with its verb. The grammar rule translator automatically translates these rules into the Prolog clauses shown earlier.

Our parser is currently being developed to analyze texts from the MEDLINE system. We have found the need to preprocess these texts before they are passed to the syntax rules and lexical lookup. The preprocessor first reads in a text and breaks it into sentence units. This is not entirely straightforward because of the high frequency of abbreviations and special punctuation in these texts. The preprocessor also recognizes and marks certain specialized constructions such as chemical names, abbreviations, acronyms, and parentheticals. An example sentence from a MEDLINE abstract illustrates the range of specialized constructions:

> The results obtained show that the repeated injection of
> DF (three times a week: 100 mg/kg each i.m.) delayed and
> diminished remarkably the urinary excretion of precursors
> and porphyrins as well as the accumulation of the latter
> in liver promoted by HCB (1 g/kg daily given by stomach
> tube).

The result of preprocessing is as follows:

```
[the,results,obtained,show,that,the,repeated,injection,of,
'DF', parenthetical([ three,times,a,week, punctuation(colonp),
number(100),mg,per,kg,each,'IM' ]),delayed,and,diminished,
remarkably,the urinary,excretion,of,precursors,and,porphyrins,
as,well,as,the,accumulation,of,the,latter,in,liver,promoted,by,
'HCB',parenthetical([number(1),g,per,kg,daily,given,by,stomach,
tube]) ].
```

Notice that acronyms and other abbreviations are in single quotes, parentheticals are marked, constructions like *mg/kg* are converted to *mg* **per** *kg,* and numbers are explicitly marked.

It is well known that no matter how comprehensive a dictionary is, there will always be words of the language that are not represented in the dictionary. This is due to a variety of factors: the missing word may be archaic and no longer in common use; the word may be a neologism which has not yet found its way into the standard dictionary; or the word may be specific to a particular domain, appearing only in a specialized dictionary. When the dictionary is being constructed for a purpose such as ours, to serve as a lexicon for parsing, all of these factors obtain, and more. There are space constraints to consider, in addition to the very real problem of developing a comprehensive lexicon in a reasonable period of time. In order to overcome some of these difficulties, we have developed a set of heuristics to use when the parser encounters an unknown word. Recall that the first step in the parsing process is the preprocessing of the text. The result of this preprocessing is that certain items are specially marked. This includes, notably, acronyms, abbreviations, and chemical terms (e.g., 2,3-dimethylpentanoate). It is likely that most of these terms will not appear in the lexicon; frequently used or well known acronyms such as DNA will, but many will not. If one of these specially marked items is not found, the parser will still be able to make a reasonable assumption about its role in the sentence. For example, if an item that has been marked as an acronym or abbreviation by the preprocessing rules has not been found in the lexicon, it will be assumed to be a noun.

In addition to the heuristics developed for special items such as acronyms and chemical terms, we have developed a set of heuristics for unrecognized words, based on methods of morphological analysis (see Byrd [2] for a somewhat different view of the role of morphological analysis in natural language understanding systems). The morphological analysis procedure attempts to recognize derivational variants of known words. Suffixes such as *-al, -ive, -ment, -ence,* are listed in the dictionary of suffixes and allow a potentially unknown word such as *procedural* to be recognized as an adjectival variant of the noun *procedure.* The suffixes are marked according to what word class types they may attach to, as well as what change they cause as part of the derivational process. For example, the suffix *-ic* is marked as attaching to a noun form in order to derive an adjectival form, e.g., *cystic* is derived from *cyst.* Further, the rules of morphological analysis attempt to recognize greco-latin root forms which are so common in the biomedical domain (See Norton [13] , Pacak [14] , and Wolff [20] for some sophisticated analyses of this type). For example, a word like *cardiopulmonary,* if not found in the main lexicon, would be checked against the lexicon of root forms, and would be analyzed as consisting of two roots: *cardi(o)* and *pulmon,* and an adjectival suffix, *-ary.*

The morphological analysis, or stemming, procedure is interleaved with the lookup procedure in order to ensure that all possible available information about a word is retrieved. Notice that if the word is completely decomposed before lookup, there is the danger of an incorrect analysis. Consider the word *melancholic.* A complete decomposition will yield: *melan* + *chol* + *ic.* The root *melan* means *dark, or black* (compare *melanin* - a dark pigment). The root *chol* means *bile* (compare *cholangitis* -

inflammation of one or more bile ducts). This is, of course, an incorrect analysis of the word as it is used today. It is of historical interest that a predominance of black bile was thought to cause sadness, but it is an incorrect analysis of its current meaning.

Lexical knowledge is crucial at all stages of the linguistic analysis: morphology, syntax, and semantics. It is needed for morphological analysis of unknown lexical items, providing the necessary roots, prefixes, and suffixes. It specifies the category and complement information that determines the syntactic structure of the sentence, and it relates the surface syntax to its logical interpretation. Finally, and importantly, the lexicon serves as a bridge to domain specific knowledge. It records the specialized vocabulary of the domain and provides links to non-linguistic knowledge bases.

References

1. Barr, A. and Feigenbaum, E. (1981). *The Handbook of Artificial Intelligence*. Los Altos: William Kaufman, Inc.

2. Byrd, R. (1983). *Word Formation in Natural Language Processing Systems.* In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*. Karlsruhe, West Germany.

3. Carbonell, J., Evans, D., Scott, D, Thomason, R. (1986). *On the Design of Biomedical Knowledge Bases.* In *Proceedings of the Fifth Conference on Medical Informatics.* Amsterdam: North-Holland.

4. Chomsky, N. (1981). *Lectures on Government and Binding.* Dordrecht: Foris.

5. *Dorland's Illustrated Medical Dictionary,* twenty fifth edition, (1974). Philadelphia: W.B. Saunders.

6. Grishman, R., & Kittredge, R. (eds) (1986). *Analyzing Language in Restricted Domains: Sublanguage Description and Processing.* New Jersey: Lawrence Erlbaum.

7. Hobbs, J.R. (1986). *Sublanguage and Knowledge* In *Grishman & Kittredge, 53-68.*

8. Kittredge, R., & Lehrberger, J. (eds) (1982). *Sublanguage: Studies of Language in Restricted Semantic Domains.* Berlin: de Gruyter.

9. Lehrberger, J. (1986). *Sublanguage Analysis.* In *Grishman & Kittredge, 19-38.*

10. Macdonald, R., Troike, R., Galvan, M., McCray, A., Shaefer,L., & Stupp, P. (1982). *Improving Techniques in Teaching English for the Job.* Rosslyn: InterAmerica Research Associates. Contract 300-80-0810, U.S. Department of Education.

11. McCord, M. (1985). *Modular Logic Grammars.* In *Proceedings of the Twenty third Annual Meeting of the Association for Computational Linguistics.* University of Chicago.

12. McCord, M. (1982). *Using Slots and Modifiers in Logic Grammars for Natural Language.* In *Artificial Intelligence 18, 327-367.*

13. Norton, L.M., & Pacak, M. (1983). *Morphosemantic Analysis of Compound Word Forms Denoting Surgical Procedures.* In *Methods of Information in Medicine, 22, 29-36.*

14. Pacak, M., Norton, L., & Dunham, G. (1980). *Morphosemantic Analysis of -itis Forms in Medical Language.* In *Methods of Information in Medicine, 19, 99-105.*

15. Partee, B. (1985). *Report of Workshop on Information and Representation.* Washington, D.C.

16. Pereira, F. & Warren, D.H.D (1980). *Definite Clause Grammars for Language Analysis- A Survey of the Formalism and a Comparison with Augmented Transition Networks.* In *Artificial Intelligence 13, 231-278.*

17. Procter, P. (ed) (1978). *Longman Dictionary of Contemporary English.* Burnt Mill: Longman Group Limited.

18. Sager, N. (1986). *Sublanguage: Linguistic Phenomenon, Computational Tool.* In Grishman & Kittredge, 1-17.

19. Sager, N (1982). *Syntactic Formatting of Science Information.* In Kittredge & Lehrberger, 9-26.

20. Wolff, S. (1984). *The Use of Morphosemantic Regularities in the Medical Vocabulary for Automatic Lexical Coding.* In *Methods of Information in Medicine, 23, 195-203.*