Computerizing a Machine Readable Dictionary

G. Jan WILMS
Computer Science
Mississippi State University
Mississippi State, MS 39762

Abstract
Current research in natural language processing is characterized by the development of theories of grammar which strongly depend on the lexicon to drive parsing systems (e.g. Lexical Function Grammar, General Phrase Structured Grammar, Functional Unification Grammar). These requirements go far beyond the typical small, hand-coded vocabularies developed for theoretical or demonstration purposes. Many researchers have independently discovered the rich, though unstructured, knowledge sources that machine readable dictionaries offer. This paper reports on an attempt to impose structure to the *Funk and Wagnalls* Dictionary, by means of a parser written in Turbo Pascal, using a mixed approach of pattern matching and transition networks. The resulting computerized dictionary is 95 % accurate, but correcting the final 5% incorrectly parsed involves painstakingly scrutinizing the output and modifying the parser to handle exceptional cases that occur only once or twice in the entire MRD, or editing the machine readable dictionary to remove errors introduced by the OCR process.

Introduction

Several researchers have recognized the usefulness of online dictionaries in many areas of natural language processing like parsing, speech generation, question answering, machine translation, etc. (for an overview, see AMSLER 84 and BOGURAEV and BRISCOE 88). Michiels distinguished early on between a machine readable dictionary (MRD), which is simply a huge ASCII text file, and a **computerized dictionary**, which has been processed to reflect the structure of the dictionary in an organized way [MICHIELS 81]. This is by no means a trivial task, as is attested by Kaplan, whose team needed about three man-years to structure the type-setting tapes of the American Heritage Dictionary [RITCHIE 87]. While a dictionary is certainly more structured than an average document, a lot of this structure is implicit. Lexicographers, after all, target their work to a human audience, which has enough intelligence to derive the structure from such clues as fonts and page layout. These clues (most of which are available in the form of type-setting codes) are essential to make the structure explicit (e.g. in the dictionary-independent format advocated by Amsler).

The *Funk and Wagnalls* Dictionary (*F&W*) was reproduced in machine readable form by Inductel, using an **optical scanner**. The final text preserves very little of the typographical information; only one font is available, and special symbols are restricted to the ones that are part of the extended ASCII set. As a result all phonetic information has been dropped. In some cases an attempt was made to recover part of the information (e.g. superscripts like ^{1} are flagged as (1), and small caps have been replaced by upper case), but most of the time the information carried by these typographical clues has been lost. Consequently, the machine readable version in its current format is of limited use to natural language applications, and transforming it to a computerized dictionary has become much harder.

The Structure of Definitions

Definition entries in the machine version of *Funk and Wagnalls* Dictionary are fairly structured (see figure 1). Each **lemma** is followed by **hyphenation information** using a character distance encoding scheme (the result of a pre-processing phase: e.g. "ab.bre.vi.a.tion’’ is listed as ‘‘abbreviation .2321. ‘‘). Next in line is **part-of-speech** (POS) information, which is occasionally absent (proper names like ‘‘Adam’’, for example, have no POS field, nor do many compound entries, like ‘‘absolute zero’’ or ‘‘absorbent cotton’’). For many lemmas, the next field lists **conjugations**: past and gerund for (irregular) verbs [*abide .1. v. a.bode or a.bid.ed, a.bid.ing*], plurals for nouns [*adieu .1. n. pl. a.dieus, Fr. a.dieux*], and comparatives and superlatives for adjectives [*able .1. adj. a.bler*, *a.blest*]. In the printed dictionary, this information is flagged by boldface type, a typographical clue which is not preserved in the MRD.

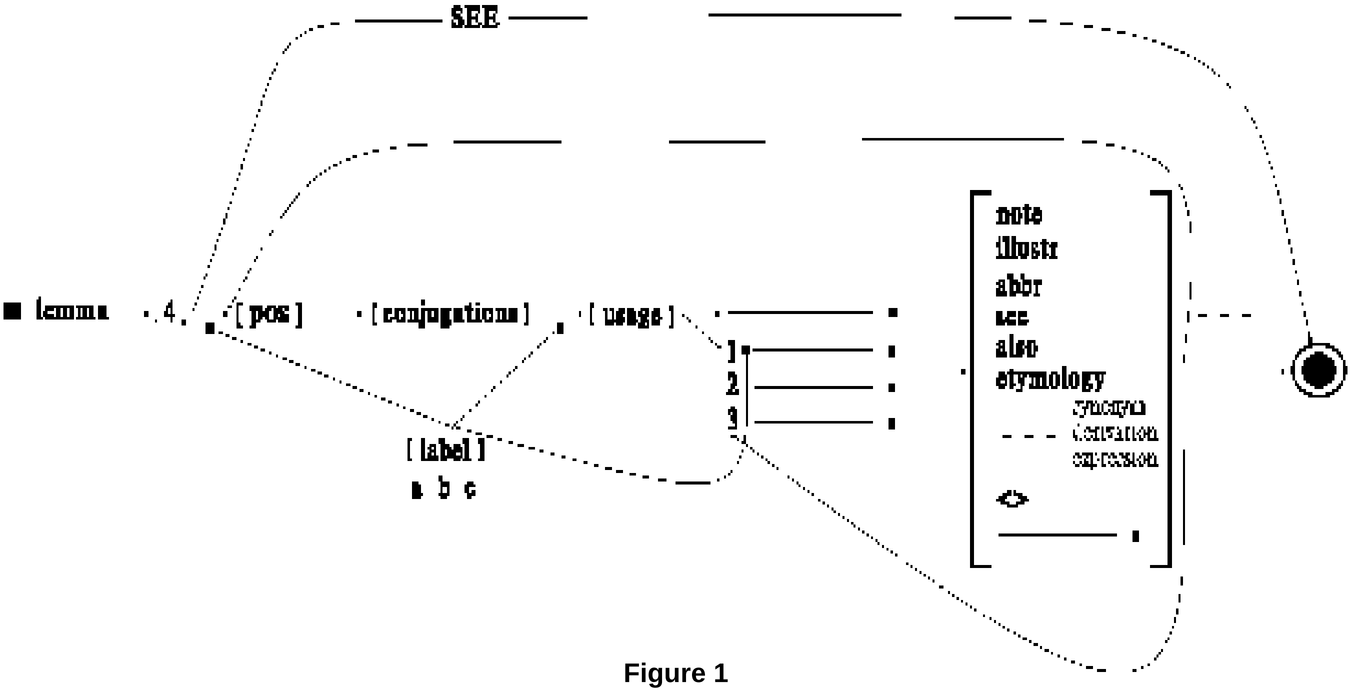

Figure 1

Many lemmas have a **label** field, either style labels (informal, dial.), currency labels (rare, archaic), language labels (Brit., Latin) and, particularly interesting to NLP, a field label which Walker calls subject codes (zool., music) [WALKER AND AMSLER 86]. Again, these are clearly marked in the printed dictionary by the use of italics, but they can occur in several locations: before the part-of-speech field [*abaft .1. Naut. adv. Toward the stern*], or following it (and optional derivations) [*arroyo .23. n. pl. .os SW U.S.*], and finally after a sense number for lemmas with multiple meanings [*a ... 3. Music a One of a series*]. As will be discussed below, this is a challenging field to parse, with many syntactical variations suggesting the idiosyncratic styles of different lexicographers.

A lemma can not only have several parts-of-speech, but each POS in turn may be subdivided into multiple meanings (senses). Most definitions are restricted to one sentence, but they may be followed by **additional information** that references that particular sense or augments the whole definition; explicit pointers to other lemma or illustrations, abbreviations, etymology, synonyms, grammatical asides, compound forms that have no separate entry, and a miscellaneous category, where anything goes, the only thing in common being that apparently one sentence was not enough to define a particular sense [*accent .2. n. ... 2. A mark used to indicate the place of accent in a word. The primary accent notes the chief stress, and the secondary accent a somewhat weaker stress. 3. ...*].

Some lemmas do not even have a definition proper; to circumvent its self-imposed restriction that definitions should ‘‘never consist of a single synonym’’ [6a], the *F&W* dictionary sometimes defines a lemma by pointing to another lemma (at times even a specific sense of the target lemma) [*acute accent .. See ACCENT (def. 3).*].

Pattern Matching

The parser takes two complimentary approaches in analyzing the structure of the dictionary. A few fields have a **limited membership** set, to which the parser has access. Part-of-speech, for example, has only 18 candidates, which moreover always occur in a predictable location of the definition. Labels too are finite, though more numerous (91 entries). An initial set was extracted from the introduction to the written dictionary, and it was expanded manually each time the parser didn’t recognize a new label correctly. As the parser covered more and more of the MRD, new entries became sparser. This list is stored in an external file and read in when the parser is initialized.

In the majority of the cases, however, **literal pattern** matching of content words cannot be used, as the number of possibilities are endless. Human readers, which are the target audience for the printed dictionary after all, can tell the fields apart with little effort, using common sense (which requires understanding of the content) and typographical clues (format). Part-of-speech, for example, stands

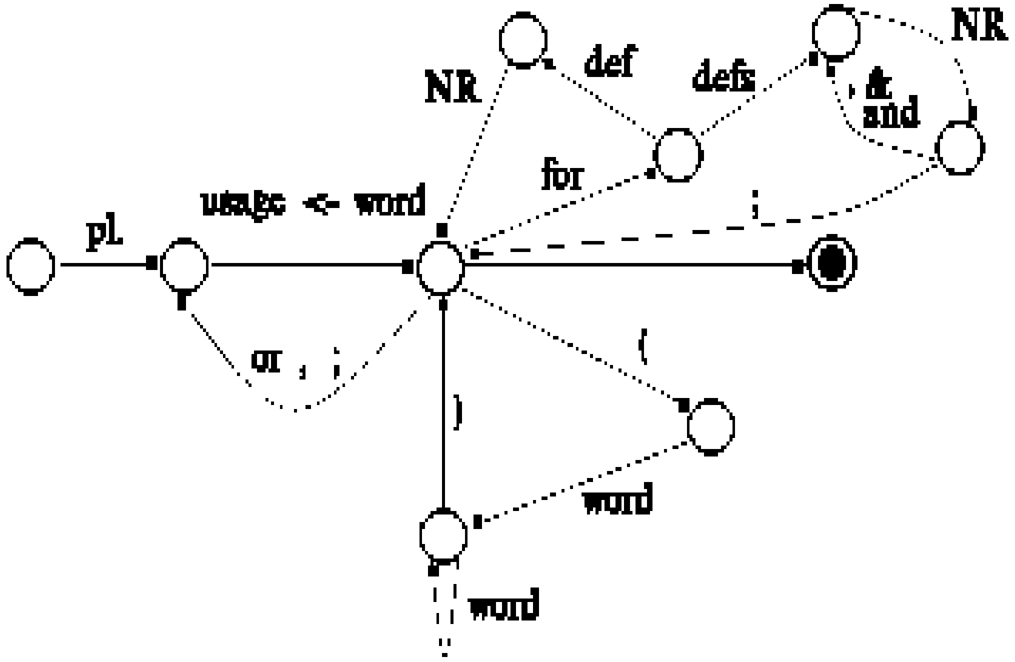

Figure 2

out because it is set in italics (but so are POS and etymology fields). As explained above, these clues are mostly lacking in the MRD.

For the majority of the fields, however, a **transition network** is used; In some instances this requires **no knowledge** about the format of the field beyond the flags that indicate its beginning and end: etymological information, for example, is always enclosed in between [ ]; a new lemma always follow■ and is terminated by hyphenation information, which facilitates packaging compound entries with embedded blanks (this was set up this way by a preprocessor).

Another example are **definition** fields that consist of **2 sentences**. These occur relatively infrequently but they don’t share any common characteristics and thus can only be identified by an elimination process: if it is not one of the fields listed in figure 1 (all of which have known patterns), it <u>must</u> be a continuation of the previous definition field.

Other important fields require some **limited knowledge** about their internal structure; as was indicated above, most definitions proper are single sentences, starting with a capital letter, and ending with a punctuation mark (period, sometimes ? or !). However, the occurrence of a period per se does not necessarily flag the end, as is the case with embedded abbreviations; [*a(5) .. Reduced var. of AB-.*]. Hence the parser has access to a list of abbreviations (which comes straight out of the appendix of the MRD). This list is even longer than the list of labels; so lengthy, in fact, that if stored in RAM in its entirety, it leaves little room for the other data structures used by the parser program. As a compromise, case is ignored [Sat. or sat.], all entries are truncated to 10 characters, and abbreviations with embedded dots [a.k.a] are eliminated.

In a few instances an abbreviation actually **does** terminate a definition [*ablative .22. adj. Gram. In some ... instrumentality, etc.*]. This is detected a posteriori, as the parser encounters the known beginning of a different field (e.g -- , or [ ), and results in controlled **backtracking**. Another example of such backtracking is when a single-word definition also happens to be a label [*analysis ... substance. 5. Psychoanalysis. [< Med. ...*].

Another case of limited knowledge that comes to play in parsing the definition proper is when a definition references a **specific sense of a hypernym**, especially of single-word definitions that supposedly don’t occur [*amylum .21. n. Starch (def. 1).*] (for a discussion of hypernyms and hyponyms, see AMSLER 81). Also embedded **example sentences are captured** [*a(2) ... for each**:** one dollar a bushel.*][*-able ... **:** eatable, solvable.*].

In some instances **detailed knowledge** of a field is required, to account for all possible variations; an example of this is the optional **CONJUGATION** field containing plural forms for nouns and past tense and gerunds for irregular verbs (see figure 2) [*bid ...* **---** *v. bade for defs. 3, 4, 6, or bid for defs. 1, 2, 5, 7, bid.den or bid, bid.ding v.t. 1. ...*]. There is no obvious flag indicating the end of this field.

Sometimes the token ‘pl.’ does not introduce a plural field, but simply indicates the current lemma is used as a plural [*acrobatics .223. n. pl. The skills ...*] or is a plural itself of another lemma [*agenda .13. n. pl. of a.gen.dum (usu. construed as sing.) A list ...*]. In these cases, this information belongs to the optional **USAGE field**, which contains miscellaneous grammatical information such as required prepositions or morphological context [*ab- ... Also: a- before m, p, v, as in avocation;*][*abstain ... refrain voluntarily: with from.*]. In order to detect these special cases, the parser uses a simple one-token lookahead mechanism, that also resolves plurals of compound lemmas which sometimes consist of multiple tokens [*agent provocateur .1822. pl. a.gents pro.vo.ca.teurs*][*able seaman .. pl. .men*].

The reason detailed information of the conjugation field is required is **ambiguity**. Though the definition proper always starts with a capital letter or a sense number, these flags may also show up in several of the fields that optionally precede it (see figure 1) [*antichrist .22. n. Often cap. A denier ...*]. Since the parser has only limited knowledge about the structure of the definition itself, correctly identifying it requires for each preceding field either an exhaustive list of all possible values (e.g. LABEL) or a mini transition network (e.g. CONJUGATIONS).

Prologizing the dictionary

The parser’s **primary purpose** is to convert the relative free-form MRD into a **structured** computerized dictionary, which can be used by natural language processing applications. Several formats have been proposed, each with specific advantages and disadvantages (see AMSLER 87 for a discussion of the merits of the SGML format), but no clear standard has emerged yet. Most researchers seem to settle for some sort of ‘‘**lispified**’’ format, in which fields are enclosed by parenthesis, and the structure is apparent from the levels of nesting. A similar approach has been adopted in this project, except that **Prolog** was chosen as a target language.

A Prolog program consists of facts and rules, the latter being Horn clauses with a single consequent followed by a list of antecedents. Symbolics Prolog makes over 1 gigabyte of RAM available for loading code + data, and clauses are indexed so that retrieval time for a random predicate is almost constant [*User’s Guide,* p 45]. Most of the information in the MRD, once identified, is easily converted to Prolog **facts**. A typical structure consists of the functor lemma, which has the following fields:

> headword, graph number, POS, list of definitions for the current POS

The default for graph number is 1, but some lemmas are homographs, i.e. while they are spelled identically, they have complete different origins and semantics (in the printed dictionary this is indicated with a superscript number). If a lemma has multiple parts of speech, there will be separate entries for each POS, unlike the printed version. Compound terms too are listed separately as a predicate [*ahead ... 3. Onward; forward.* **---** *ahead of In advance of, as in time, rank, achievement, etc.* **---** *to be ahead U.S. Informal To have as profit or advantage; be winning.* **---** *to get ahead To make one’s way socially, financially, etc.*].

Each POS of a lemma consists of one or more senses, which are collected in a list (in Prolog indicated with []). Lemmas with multiple senses are uniquely identified with a sense number which the printed dictionary uses for cross-referencing [*Automatic ... Also (except for def. 4) au’to.mat’i.cal, au.tom.a.tous.*]. Lemmas with only one sense are assigned a default number of 1 by the parser. Next follows a **LABEL field**, which is an empty list in most cases, as the field is optional. In some instances a label further subdivides a definition, as in the following example (more detailed examples can be found in appendix 1):

```
ablution .22. n. 1. A washing or cleansing of the body; a
bath. 2. Eccl. a A ceremonial washing of the priest’s hands or
of the chalice and paten during the Eucharist. b The liquid
used for this. [< L < ab- away + luere to wash] ---
ab.lu’tion.ar’y adj.

lemma(‘ablution’,1,n,[def(1,[],’’,[‘a washing or cleansing of
the body’,’a bath’],[]), def(2,[‘Eccl.’],’’,[‘a ceremonial wash-
ing of the priest’s hands or of the chalice and paten during
the Eucharist’],[]), def(2,[‘Eccl.’],’’,[‘the liquid used for
this’],[])]).
hyphenate(‘ablution’,1,22).
hyphenate(‘ablutionary’,1,22421).
etymology(‘ablution’,1,’< L < ab- away + luere to wash’).
```

The actual DEFINITION field is preceded by an optional USAGE field and followed by an optional **EXAMPLE field**. Examples are detected by the parser by virtue of the fact that the MRD flags them with a ‘**:**’. That punctuation mark is sometimes used in another context, though [*abandon ... 2. To surrender or give over: with to.*], so the parser targets only sentences that repeat the lemma [*able ... 2. Having or exhibiting superior abilities; skillful: an able writer.*]. Isolating these examples allows a NLP program to capture some of the common sense and world knowledge that is implicit in the dictionary (see JENSEN and BINOT 87 and 88).

Some of the information in the MRD is turned into **Prolog**

rules, primarily to avoid duplicating unnecessary information. The theorem prover around which every Prolog compiler is built will automatically attempt to satisfy the antecedent of the rule, and because of the unification principle, the left hand side of the rule will share the definition that was returned; the following example illustrates the process:

```
a(2) .. indefinite article or adj. In each; to each; for each: one
dollar a bushel. [OE on, an in, on, at]
lemma(‘a’,2,indefinite_article,[def(1,[],’’,[‘in each’,’to
each’,’for each’],[‘one dollar a bushel’])]).
lemma(‘a’,2,adj,Def) :- lemma(‘a’,2,indefinite_article,Def).
```

The same principle is applied to spelling variations [*abaca ... used for cordage. Also ab’a.ka.*].

A general principle in converting a MRD to a structured knowledge base is **never to throw any information away**, no matter how redundant or useless it may seem at the moment [BOGURAEV and BRISCOE 87]. Therefore the parser introduces several other predicates like **hyphenate**, **abbreviation**, **plural**, **root** [*abandon ...* **---** *a.ban’don.er n.* **---** *a.ban’.don.ment n.*]. Two examples of fields that are captured without much parsing, for possible later use, are **etymology** (enclosed in **[]**) and occasional lengthy **grammatical asides**, which are introduced by a <> flag.

The parser also creates a **SYNONYM** predicate, which corresponds to information found in the MRD in various places. Some obvious places are explicit references [*appraise ...* **---** *Syn. 1. evaluate, value, assess, assay.*], which at times evolve into lengthy discussions of subtle semantic nuances between related words. Another category in which the *F&W* prides itself are ‘collateral adjectives’, which are ‘‘adjectival forms of the noun so remote in spelling that they may not be brought to mind’’ [p 7a] [*arm .. n. 1. Anat. ... <> Collateral adjective: brachial.*]. Other sources are supposedly non-existent one-word definitions, which often include a pointer to a specific sense of a synonym [*amylum .21. n. Starch (def. 1). [< L < Gk.]*], and references buried toward the end of the definition text [*abomasum ...digestive stomach of a ruminant: also called reed.*].

Computational Lexicography

A by-product of the parser is that it **flags errors and inconsistencies** in the MRD. These subtle inconsistencies often go unnoticed by the human proofreader (and user), who is better at detecting semantic errors (this complementary relationship has also been noticed by Kazman, who wrote a transducer for the OED [KAZMAN 86]). A dictionary is rarely the product of a single lexicographer, and subtle differences in style are almost unavoidable (eg sometimes a USAGE field contains ‘usu. pl.’, sometimes it is spelled out ‘usually pl.’). Thus both computational linguists and lexicographers can benefit from working together; the former get access to an enormous repository of syntactic and semantic knowledge, the latter can achieve better consistency from computerized assistance.

Some of these errors actually resulted from the **optical-scanning process** by which the *F&W* was turned into a MRD. These errors are often critical, especially when they happen to involve flags that drive the parser [pt. iso pl. in austerity, or vt. iso v.t. in affect(2)]. A missing or incorrect sense number [2O iso 20 in cast] would barely be noticed by a human reader, but it confuses the parser enough that it is forced into **panic mode**, where it progressively deletes a word from the input string until it encounters a flag it recognizes, or until the beginning of the next lemma is reached. Since some of these errors occur relatively frequently, enough knowledge has been built into the parser to perform auto-recovery, and to continue processing after logging the type and location of the error. This allows the manual updating of the MRD source in batch mode (as far as the parser is concerned, this step is even redundant). An example are the m-dashes (**---**) which often look like **--** or **----** [*allay to lay* **--** *al.lay’er n.*], or are concatenated to the previous token [abase].

Debugging facilities

Writing a parser for a textbase of approximately 9 meg may be less challenging than analyzing free-format natural language text, but it is still a continuously evolving project. While it is fairly easy to cover 95 % of the input text, the law of **diminishing returns** is obvious in trying to add code to correctly parse the **remaining 5 %**. Some of this is caused by new errors that are constantly found in the MRD, errors too infrequent to accumulate knowledge for autorecovery, and hence human intervention is necessary to correct the MRD source. Some of the remainder, however, are not errors, but things that occur only once or twice in the whole dictionary, and are thus unanticipated by the parser. An example is [*atto- ... quintillionth [10 (to the negative ...*], where ‘**[**‘ does not flag an etymology field, and [*adieu .1. n. pl. a.dieus, Fr. a.dieux A farewell.*], where the label Fr. came unexpectedly. As a result, it is **not** expected that this situation will **improve** as the parser progresses, in contrast to the CYK project, which anticipates an increasing rate of automation after the first 1% has been hand processed [LENAT et al. 86].

To facilitate the debugging process, several **tools** have been built into the parser. There is a facility that allows the programmer to quickly **zoom in** on a particular lemma, without having to process all the lemmas that precede it. As the parser progresses, it displays the current lemma on the screen, and a **progress bar** at the bottom of the screen indicates how much of the operation has been completed. This is possible because the parser knows how many lemmas each of the 26 subdictionaries (lemmas starting with A, B, ..Z) contain. As was mentioned earlier, the parser also collects information about ambiguous cases into an external **log file** with enough context information so the programmer does not have to (solely) rely on closely analyzing the parsed output for possible bugs. An example are messages indicating that the parser had to backtrack, because these often suggest that the period that flags the end of a definition field was missing [*accordion ... a bellows operated by the performer [< Ital. ...*].

Closing Observations

The **planned use** of the MRD is to assist the natural language front-end to a knowledge system that automatically extracts knowledge for machine readable reference sources (in particular from the *Merck Veterinary Manual*). In parsing the manual, the NL processor has access to both the domain specific knowledge stored (and growing) in the knowledge base, and to the MRD for more general syntactic, semantic, and pragmatic information. Having all this information available may have an adverse affect, however, as the number of possible parses increases! This results from the fact that there is ‘‘no significance to [the] order’’ of the senses or the POS of a lemma in the MRD; ‘‘the first definition is not necessarily for the earliest use, nor is it the most frequently used’’ [*F&W* p. 6a]. As Kucera puts it, ‘‘the problem is not simply tagging by statistical analysis, but rather some form of usage determination based on a corpus of current citations’’ [KUCERA 85]. An additional field is proposed for each sense of a lemma, with a weight that ranks it with respect to the domain of study (veterinary medicine), using the association ratio measure proposed by Church [CHURCH 89] (folds, for example has 9 senses as a verb, 6 as a noun, and the only thing that currently might identify the 5th sense of the noun as the correct one (skin folds) is the label anat.). With domain specific ranking the number of possible parses could be reduced, or ranked in order of suitability to the context of the domain of study.

Selected Bibliography

Amsler R A (1981) A Taxonomy for English Nouns and Verbs. *Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics*, Stanford, California, pp 133-8

Amsler R A (1984) Machine-readable dictionaries. In M E Williams (ed) *Annual Review of Information Science and Technology (ARIST).* American Society for Information Science, Vol.19, pp 161-209

Amsler R A, Tompa F W (1987) *An SGML-based Standard for English Monolingual Dictionaries*

Boguraev B K, Briscoe E J (1987) Large Lexicons for Natural Language Processing: Exploiting the Grammar Coding System of LDOCE. *Computational Linguistics* 13(13): 203-218

Boguraev B K, Briscoe E J (eds) (1988) *Computational Lexicography for Natural language Processing* Longman Ltd, Harlow

Church K (1989) Word Association Norms, Mutual Information, and Lexicography *Association for Computational Linguistics* pp 76-83

Funk and Wagnalls Standard Desk Dictionary (1984) Harper and Row

Jensen K, Binot J-L (1987) Disambiguating Prepositional Phrase Attachments by Using On-Line Dictionary Definitions. *Computational Linguistics* 13(3-4): 251-260

Jensen K, Binot J-L (1988) Dictionary Text Entries as a Source of Knowledge for Syntactic and other Disambiguations 2nd Conference on Applied Natural Language Processing, *ACL, Proceedings of the Conference,* Texas, pp 152-159

Kazman R (1986) *Structuring the text of the Oxford English Dictionary through Finite State Transduction.* Technical Report TR-86-20, Department of Computer Science, University of Waterloo, Ontario

Kucera H (1985) Uses of On-Line Lexicons. *Proceedings of the First Conference on Information in Data*, University of Waterloo Centre for the New Oxford English Dictionary, Waterloo, Canada, pp 7-10

Lenat D, Prakash M, Spepherd M (1986) CYK: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks *AI Magazine* 6(4) pp 65-85

Michiels A (1982) *Exploiting a Large Dictionary Database*. Doctoral thesis, Université de Liège, Belgium

Ritchie G (1987) Discussion Session on the Lexicon. In Whitelock et al. (eds) *Linguistic and Computer Applications.* Academic Press, London, pp 225-56

User’s Guide to Symbolics Prolog (1989)

Walker D, Amsler R (1986) The Use of Machine-Readable

Dictionaries in Sublanguage Analysis. In Grishman R, Kittredge R (eds) *Analyzing Language in Restricted Domains*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, pp 69-83

Wilms G J (1988) *Machine Readable Dictionaries and Natural Language Processing* Unpublished Manuscript

Wilms G J (1989) *Generating a Concordance of One-Word Definitions Using a Computerized Dictionary* Unpublished Manuscript

Appendix 1 Sample Entries

a .. n. pl. a’s or as, A’s or As, aes 1. The first letter of the English alphabet. 2. Any sound represented by the letter a. -- symbol 1. Primacy in class. 2. A substitute for the numeral 1. 3. Music a One of a series of tones, the sixth in the natural diatonic scale of C, or the first note in the related minor scale. b A written note representing this tone. c The scale built upon A. 4. Chem. Argon (symbol A).

a(2) .. indefinite article or adj. In each; to each; for each: one dollar a bushel. [OE on, an in, on, at]

a(3) .. indefinite article or adj. One; any; some; each: expressing singleness, unity, etc., more or less indefinitely. It is used: 1. Before a noun expressing an individual object or idea: a bird; a hope. 2. Before an abstract noun used concretely: to show a kindness. 3. Before a collective noun: a crowd. 4. Before a proper noun denoting a type: He is a Hercules in strength. 5. Before plural nouns with few, great many, or good many: a few books. 6. After on, at, or of, denoting oneness, sameness: birds of a feather. <> Before vowel sounds the form becomes an. See note under AN. [Reduced form of AN used before consonant sounds]

a- .. prefix In; on; at: aboard, asleep, agog, agoing. [OE on, an in, on, at]
a-(2) .. prefix Up; on; away: arise, abide. [OE a- up, on, away]
a-(3) .. prefix Of; from: athirst, akin, anew. [OE of off, of]
a-(4) .. prefix 1. Without; not: achromatic. 2. Apart from; unconcerned with: amoral. [Reduced form of AN-]
a-(5) .. Reduced var. of AB-.
a-(6) .. Reduced var. of AD-.

```
lemma(‘a’,1,n,[  def(1,[],’’,[‘the first letter of the English alphabet’],[]),
                 def(2,[],’’,[‘any sound represented by the letter a’],[])]).
lemma(‘a’,1,symbol,[
        def(1,[],’’,[‘primacy in class’],[]),
        def(2,[],’’,[‘a substitute for the numeral 1’],[]),
        def(3,[‘Music’],’’,[‘one of a series of tones, the sixth in the natural diatonic scale of C, or the first note in the related
minor scale’],[]),
        def(3,[‘Music’],’’,[‘a written note representing this tone’],[]),
        def(3,[‘Music’],’’,[‘the scale built upon A’],[]),
        def(4,[‘Chem.’],’’,[‘argon (symbol A)’],[])]).
lemma(‘a’,2,indefinite_article,[def(1,[],’’,[‘in each’,’to each’,’for each’],[‘one dollar a bushel’])]).
lemma(‘a’,3,indefinite_article,[
        def(1,[],’’,[‘one’,’any’,’some’,’each’,’expressing singleness, unity, etc., more or less indefinitely’],[]),
        def(1,[],’’,[‘it is used before a noun expressing an individual object or idea’],[‘a bird’,’a hope’]),
        def(2,[],’’,[‘it is used before an abstract noun used concretely’],[‘to show a kindness’]),
        def(3,[],’’,[‘it is used before a collective noun’],[‘a crowd’]),
        def(4,[],’’,[‘it is used before a proper noun denoting a type’],[‘he is a Hercules in strength’]),
        def(5,[],’’,[‘it is used before plural nouns with few, great many, or good many’],[‘a few books’]),
        def(6,[],’’,[‘it is used after on, at, or of, denoting oneness, sameness’],[‘birds of a feather’])]).
lemma(‘a-’,1,prefix,[def(1,[],’’,[‘in’,’on’,’at’],[‘aboard’,’asleep’,’agog’,’agoing’])]).
lemma(‘a-’,2,prefix,[def(1,[],’’,[‘up’,’on’,’away’],[‘arise’,’abide’])]).
lemma(‘a-’,3,prefix,[def(1,[],’’,[‘of’,’from’],[‘athirst’,’akin’,’anew’])]).
lemma(‘a-’,4,prefix,[  def(1,[],’’,[‘without’,’not’],[‘achromatic’]),
                def(2,[],’’,[‘apart from’,’unconcerned with’],[‘amoral’])]).
lemma(‘a-’,5,_,[def(1,[],’’,[‘reduced var. of AB-’],[])]).
lemma(‘a-’,6,_,[def(1,[],’’,[‘reduced var. of AD-’],[])]).

lemma(‘a’,2,adj,Def) :- lemma(‘a’,2,indefinite_article,Def).
lemma(‘a’,3,adj,Def) :- lemma(‘a’,3,indefinite_article,Def).
lemma(‘a-’,0,prefix,[def(0,Label,’before m, p, v’,Def,[‘avocation’])]) :- lemma(‘ab-’,1,prefix,[def(1,Label,_,Def,_)]).

grammar(‘a’,3,indefinite_article,’Before vowel sounds the form becomes an.’).
```

```
see(‘a’,3,indefinite_article,’an’,1,_).

etymology(‘a’,2,’OE on, an in, on, at’).
etymology(‘a’,3,’Reduced form of AN used before consonant sounds’).
etymology(‘a-’,1,’OE on, an in, on, at’).
etymology(‘a-’,2,’OE a- up, on, away’).
etymology(‘a-’,3,’OE of off, of’).
etymology(‘a-’,4,’Reduced form of AN-’).

hyphenate(‘a’,1,0).
hyphenate(‘A’,1,0).
hyphenate(‘a’,2,0).
hyphenate(‘a’,3,0).
hyphenate(‘a-’,1,0).
hyphenate(‘a-’,2,0).
hyphenate(‘a-’,3,0).
hyphenate(‘a-’,4,0).
hyphenate(‘a-’,5,0).
hyphenate(‘a-’,6,0).

plural(‘a’,1,’a’s’).
plural(‘a’,1,’as’).
plural(‘a’,1,’A’s’).
plural(‘a’,1,’As’).
plural(‘a’,1,’aes’).
```

Appendix 2 Debug Sample

```
0085 magic                TOO FAR             : ...beautiful
0087 magician             TOO FAR             : ...legerdemai
0308 manes                CHECK_FOR_POS       : CONCATENATED n.pl.
0326 -mania               PANIC               : In
0336 manifold             --                  : too long/short ----
0362 manor                PANIC               : b
0390 Maoism               CHECK_FOR_POS       : CONCATENATED n.,
0390 Maoism               PANIC               : ,
0414 marchese             LOCATE POS          : concattenated n.fem.
0458 marksman             LOCATE POS          : concattenated n.fem.
0516 Marxism-Leninism     CHECK_FOR_POS       : CONCATENATED n.,
0516 Marxism-Leninism     PANIC               : ,
0545 massage              LOCATE POS          : concattenated n.fem.
0546 masseur              LOCATE POS          : concattenated n.fem.
0598 mate                 FIND USAGE          : missing , ed,
0612 mathematics          CHECK_FOR_POS       : CONCATENATED n.pl.
0661 May                  ETYMOLOGY           : CONCATENATED May]NOTE  :
0711 measles              CHECK_FOR_POS       : CONCATENATED n.pl.
0720 meat packing         TOO FAR             : ...meat packer
0727 mechanics            CHECK_FOR_POS       : CONCATENATED n.pl.
0771 mediterranean        ETYMOLOGY           : CONCATENATED earth]NOTE  :
0810 meliorate            TOO FAR             : ...ameliorate
0820 melodramatics        CHECK_FOR_POS       : CONCATENATED n.pl.
0837 memorabilia          CHECK_FOR_POS       : CONCATENATED n.pl.
```