Extracting Representation from Structured Text: Initial Steps

Robin Drogemuller[1], Rob Woodbury[2], John Crawford[1]
1 CSIRO, Building, Construction and Engineering
2 University of Adelaide, School of Architecture, Landscape Architecture and Urban Design

ABSTRACT: A great deal of work has been done in the past on natural language recognition within the field of artificial intelligence. The aim of this work was to allow natural language text to be read in by a computer and structured in a format that would allow automatic interpretation of the text. This was intended to reduce the "knowledge engineering bottleneck" that has been a significant constraint on the use of artificial intelligence techniques within many fields. Some similar work has also been done within the AEC industry concentrating mainly on building codes.
The research project described in this paper aims to simplify the analysis of structured text and its conversion into computer interpretable forms by providing support with computer software. The work is built around two documents – a glossary of building terms used in Australia and the Building Code of Australia. The various issues concerned with "noise" in the source data, the structure and content of documents to be analysed and the desired computer interpretable result will be presented.
This work is motivated by:

- *the need to maintain BCAider, a knowledge based system that assists in checking building designs for compliance with the Building Code of Australia;*
- *continuing work in encoding of regulations in computer interpretable form; and*
- *the need for international glossaries to support information harmonisation efforts such as the IAI and STEP.*

The software suite under development assists people with some understanding of language structure and knowledge engineering in converting structured text into computer interpretable form using a visual user interface. The current state of the design and development of this software suite will be described and the results of its use presented.

KEYWORDS: Building Codes, Natural Language Analysis, Parsing

1 MOTIVATION

The development of automatic or semi-automatic methods of computer interpretation of codes and standards has been a long held goal by various groups.
CSIRO has an existing commercial product, BCAider (CSIRO, 1999), which is a "classic" knowledge based system that assists a user to check a building design for compliance against the Building Code of Australia (BCA) (ABCB, 1996). BCAider consists of an inference engine specialised to handle tables and images an addition to rules and a separate rules base. The rules base is represented in a format that was based on the KnowledgePro format, with extensions required to support the additional user interface requirements.
Due to 6 monthly modifications to the BCA, BCAider needs to be updated regularly. Work has been underway for some time on integrating BCAider with architectural CAD systems. The BCAider engine has been used to provide checking against the Malaysian building code and work has been started on a new Indonesian code. There is also a long term ambition to use the BCAider engine to allow checking against other codes. Currently, all of the knowledge engineering is done by hand. This means that over 5 person years of effort is required to encode an appropriately structured, moderately sized building code.
These factors have lead to the current work on semi-automating the knowledge elicitation and knowledge engineering of building codes.

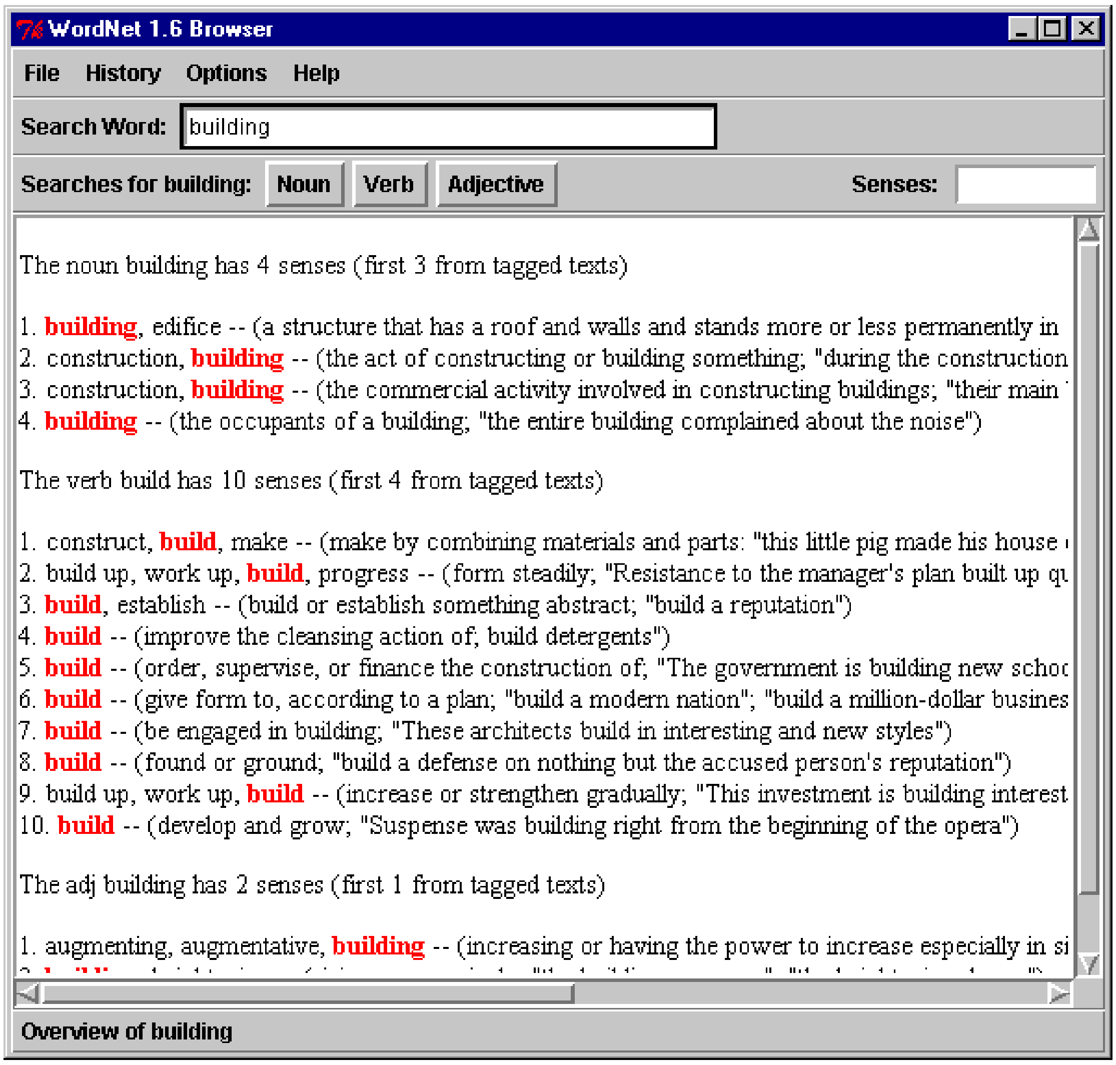
2 EXISTING WORK IN TEXTUAL ANALYSIS

There has been nearly forty years of active research into the computer analysis and interpretation of natural language and text. Much of the initial impetus came from the need of the US military for translation of technical work into English. There have been significant achievements that can provide a basis for building code work.
When interpreting natural language text as a text file there are a number of steps that need to be followed:

1. Breaking the text into tokens (words and punctuation);
2. Syntactic analysis (parsing) to identify the structure of the sentence(s);
3. Semantic interpretation to build the internal representation of the text.

Tokenising text is relatively trivial. The only significant problems lie in interpreting punctuation and in fixing "dirty" text where spurious spaces, for example, can make automatic recognition of words difficult.
Syntactic analysis requires that the grammatical type of each word (noun, verb, adjective, etc) is tagged and the appropriate sentence structure identified against the grammar defined within the parser. Derived forms of words (ie "s" for plural) also need to be identified. The appropriate senses of each word must be identified and the words and grammatical structures disambiguated. For example, WordNet (Fellbaum, 1998) identifies 16 senses of the word

"building" - 4 noun senses, 10 senses of the verb and 2 adjectival senses. Obviously, when interpreting text all sentences must be recognised, so some human intervention is required when building up the grammar.
Semantic interpretation involves the conversion of the parser output (parse tree) into a logical representation. Russell and Norvig (1995) note that intermediate forms that lie between parse trees and logical representations are often used to ensure that all variables in the logical sentences can be properly resolved.
Other problems need attention when interpreting text, such as indexicals (pronouns and time references) and anaphora (references to objects mentioned previously). Another problem that requires human intervention is when domain knowledge outside of the text is required to understand the text.



Search Word: building

Searches for building:

The noun building has 4 senses (first 3 from tagged texts)

1. **building**, edifice -- (a structure that has a roof and walls and stands more or less permanently in
2. construction, **building** -- (the act of constructing or building something; "during the construction
3. construction, **building** -- (the commercial activity involved in constructing buildings; "their main
4. **building** -- (the occupants of a building; "the entire building complained about the noise")

The verb build has 10 senses (first 4 from tagged texts)

1. construct, **build**, make -- (make by combining materials and parts: "this little pig made his house
2. build up, work up, **build**, progress -- (form steadily; "Resistance to the manager's plan built up qu
3. **build**, establish -- (build or establish something abstract; "build a reputation")
4. **build** -- (improve the cleansing action of; build detergents")
5. **build** -- (order, supervise, or finance the construction of; "The government is building new schoo
6. **build** -- (give form to, according to a plan; "build a modern nation"; "build a million-dollar busines
7. **build** -- (be engaged in building; "These architects build in interesting and new styles")
8. **build** -- (found or ground; "build a defense on nothing but the accused person's reputation")
9. build up, work up, **build** -- (increase or strengthen gradually; "This investment is building interest
10. **build** -- (develop and grow; "Suspense was building right from the beginning of the opera")

The adj building has 2 senses (first 1 from tagged texts)

1. augmenting, augmentative, **building** -- (increasing or having the power to increase especially in si
[illegible]

Figure 1. WordNet Graphical User Interface

The extensive work on natural language understanding provides a range of useful tools. Machine tractable dictionaries, lexicons and parsers that tag tokens and identify grammatical structure according to various theories are widely available. (See http for listing of resources) WordNet and the Link Grammar Parsing System have been used in the work described in this paper.
WordNet (Fellbaum, 1998; http://www.cogsci.princeton.edu/~wn/) is a lexical database that stores nouns, adjectives, adverbs and verbs. These are linked using semantic networks into sets of synonyms (synsets) and relations between these synsets. Each synset clusters the words that can be used to express a single concept. Synsets are related through the following relations:

- superordinate/subordinate (nouns),
- part-whole (nouns),
- opposition (antonyms),
- manner (verbs) and
- pertainymy (between nonpredicate adjectives and particular nouns).

The WordNet database can be accessed either through a graphical interface (figure 1) or a command line interface. In this work the command line interface is used to access the WordNet database from within the parsing software.
It should be noted that WordNet does not attempt to cover all of the words in the English language. Pronouns, conjunctions, etc are not stored in the Wordnet database.

```
linkparser> The objective of this section is to safeguard occupants from illness or
injury while evacuating in an emergency.

No complete linkages found.
Found 20 linkages (20 had no P.P. violations) at null count 1
  Linkage 1, cost vector = (UNUSED=1 DIS=4 AND=0 LEN=38)

  +-------------------------------------------------------------------Xp-----
  |            +------------Ss------------+          +-----------------------
  +-----Wd-----+       +----Js----+       |          +---------MVp--------+--
  |    +---Ds--+---Mp--+   +--Dsu-+       +-TO+---I--+-----Op----+        |
  |    |       |       |   |      |       |   |      |           |        |
///// the objective.n of this section.n is.v to safeguard.v occupants.n from


-------------------------------------------------------------+
---------MVp--------------------------------+                |
-----------------J------------------+       +----Jp---+      |
    +---------------AN--------------+       |  +--Dsu-+      |
    |                               |       |  |      |      |
illness.n or injury.n [while] evacuating.g in an emergency.n .


  +-------------------------------------------------------------------Xp-----
  |            +------------Ss------------+          +-----------------------
  +-----Wd-----+       +----Js----+       |          +---------MVp--------+--
  |    +---Ds--+---Mp--+   +--Dsu-+       +-TO+---I--+-----Op----+        |
  |    |       |       |   |      |       |   |      |           |        |
///// the objective.n of this section.n is.v to safeguard.v occupants.n from


-------------------------------------------------------------+
---------MVp--------------------------------+                |
-----------------J------------------+       +----Jp---+      |
                 +--------AN--------+       |  +--Dsu-+      |
                 |                  |       |  |      |      |
illness.n or injury.n [while] evacuating.g in an emergency.n .

Press RETURN for the next linkage.
```

Figure 2. Link Grammar Parsing System Analysis

The Link Grammar Parsing System (http://www.link.cs.cmu.edu/link) uses link grammar theory to identify the linkages between words (Figure 2). These linkages resolve the problems of indexicals and anaphora discussed previously. The output is a tree describing the structure of the sentence. Tags are also placed on identified words.
The resultant link trees are not unique. Figure 2 illustrates two of the resultant trees. Detailed explanations of the linkages are given at the URL. As noted below, clauses within a code should be unambiguous in their interpretation. As this analysis shows, Objective DO1 (b) is ambiguous in that domain knowledge is required to eliminate the possible interpretation "safeguard occupants from illness while evacuating in an emergency, or safeguard occupants from injury while evacuating in an emergency".

3 CODES AND STANDARDS

The computer interpretation of codes and standards is not as complex as "normal" natural language. In a well written code:

- the grammar will be reasonably consistent;
- the vocabulary will be of a restricted scope;
- the number of senses in which specific words can be used are reduced;
- the context of clauses is unambiguous
- the meaning of clauses is unambiguous;
- indexicals and anaphora (see above)) are restricted to within a clause;
- logically consistent clauses;
- the clauses are typed in the sense that they follow patterns.

Aspects of codes that make the interpretation task more difficult are:

- the use of clause numbering systems;
- longer sentences with more complex grammar;
- the consequences of incorrect interpretations.

The previous work on implementing BCAider (*ref*)was done in parallel with the writing of the BCA. The formal approaches required during knowledge engineering identified and lead to the resolution of areas of ambiguity in the wording of the clauses.

4 PREVIOUS WORK ON INTERPRETATION OF CODES AND STANDARDS

The area of *interpretation of codes and standards* (more generally called *standards processing*) has long been divided into two parts, one concerned with representation, the other classification. When they are considered together, a hybrid formalism is the usual result, for example, Garrett and Hakim's (1992) object-oriented model that comprises several distinct hierarchies and Yabuki and Law's (1993) separation of representation by objects versus classification by logic. In the representation arena, ideas of creating highly structured textual representations of codes were amongst the earliest mooted. Much work has been done on the extraction of structured information from written codes, notably by national research bodies associated with groups responsible for codes, for example, that of the Minicode Generator (Vanier 1995) and hypermedia-based representation such as that reported by Turk and Janez(1995). In Australia (ref) and other places (ref) this co-location of code researchers and writers has created a feedback process that has resulted in new versions of the relevant code being more amenable to computer-based representation. This fact of history places the area into light very different from early researchers who had to struggle with less symbolically-oriented codes (Fenves 1987,Rosenman and Gero 1985).

5 THE BUILDING CODE OF AUSTRALIA

The Building Code of Australia (BCA) is written as a performance specification. It consists of two volumes, one covering single family residential buildings and non-habitable structures and the other volume covering all other building types.

Each Section covers a different aspect or "system" within buildings. Within each section Objectives, Functional Statements, Performance Requirements and Deemed-to-Satisfy provisions are given for the "system". The Objectives and Functional Statements describe the overall intentions of the Section and are too "open" or general to provide an unambiguous method of determining compliance of a building. The Performance Requirements identify the requirements that must be met without providing specific metrics for assessment. This level provides the flexibility required for modern codes so that innovation is still possible. The Deemed-to-Satisfy provisions provide "cook book" solutions which have been assessed as meeting the Performance Requirements. The Deemed-to-Satisfy provisions provide an explicit metric through an absolute statement that a particular approach complies. They also provide an implicit metric, since under the "expert assessment" provisions of the BCA they can be used as a basis for comparison of a proposed innovative solution with an acknowledged acceptable solution.

The hierarchy of detail allows clear separation of "open" and "closed" concepts. "Open" concepts are those that are not computer interpretetable. "Closed" concepts are computer interpretable since they use concepts that can represented in a object-oriented CAD system.

Section D Access and Egress
DO1
The Objective of this Section is to-

…

(b) safeguard occupants from illness or injury while evacuating in an emergency.

DF2
A building is to be provided with means of evacuation which allow occupants time to evacuate safely without being overcome by the effects of an emergency.

DP4
Exits must be provided from a building to allow occupants to evacuate safely, with their number, location and dimensions being appropriate to-

(a) the travel distance; and
(b) the number, mobility and other characteristics of occupants; and
(c) the function or use of the building; and
(d) the height of the building; and
(e) whether the exit is from above or below ground level.

Clause: D1.2 Number of exits required

(a) All buildings - Every building must have at least one exit from each storey.
(b) Class 2 to 8 buildings - In addition to any horizontal exit, not less than 2 exits must be provided from the following:
  (i) Each storey if the building has an effective height of more than 25 m.

…

Figure 3. Partial extract from Building Code of Australia

The clauses selected in Figure 3 were chosen to illustrate the hierarchical structure of the BCA, where clause D1.2 provides a deemed-to-satisfy solution for Performance Requirement DP4. DP4 is the Performance Requirement for

Functional Statement DF2, which relates to part of Objective DO1 (b). The letter "D" in front of each identifier indicates that all of these are extracted from Section D of the BCA.

6 TEXTUAL ANALYSIS FOR CODE INTERPRETATION

Textual analysis of codes requires some form of dictionary, a grammar that enables identification of the structure of clauses and a mapping between the grammar and an internal representation. Currently these are in various stages of development.

The dictionary is being built up using a simple program that reads the text of the BCA one word at a time. Each word is read and checked against the existing internal database. If the word is already in the database, the next word is read. If the word is not in the database, it is looked up in WordNet. If the word is in the WordNet database it is added to the dictionary, along with it possible uses (ie, noun, adjective, verb, etc). If the word is not in the WordNet database the uses are entered by hand from a printed dictionary. The dictionary building program does not currently use the tagging capabilities of the Link Grammar software, although this will be added in the future.
The various forms of words, such as plurals of nouns and participles of verbs need special attention. The plural forms of nouns which do not add "s" to indicate the plural can be stored as exceptions. While processing text, any noun that does not have a specific plural assigned is then assumed to take either "s" or "es" depending on the final letters of the word. Currently, the participles of verbs are all explicitly stored in the dictionary with a link to the infinitive form.
The grammar rules are being built up using the Link Grammar Parsing System. The process involves manual removal of clause numbers and the breakdown of sentences into less complex sentences at conjunctions (and) and disjunctions (or) before feeding the sentence into the Link Grammar Parsing System. The results are then disambiguated by manual selection of the appropriate tagged sentence. Extra tags are also manually inserted where the Link Grammar Pasrser does not recognise the word. The tagged sentence is then read by a program which extracts the grammar and converts it to a definite clause grammar (DCG) structure (Figure 5). The DCG structure is then matched against existing DCG structures to build up a grammar that will allow full interpretation of the target code.

Some form of DCG rules will provide the basis for conversion of the code text to the internal representation. A number of different approaches are being tested to determine the most suitable approach.
The first method under test is the encoding of the sentences using DCGs to represent the phrasal structure of the sentence. Arguments can be added to the DCG rules to allow automatic generation of the parse tree. This can then be converted into the internal representation.
Conversion into the internal representation requires a finer distinction between the various types of words than given in dictionaries to ensure appropriate interpretation of the sentence. For example, in Objective DO1(b) (Figure 3) the word "or" is used to add two situations to the goal of "safeguarding occupants". This is the use of "inclusive or" which needs to be interpreted as meaning that both circumstances are applicable. This usage must be distinguished from the "exclusive or", ie "either one or the other". This is not surprising given the clear distinction made between the two uses of "or" in logic. Dictionaries also give "conjunction" as allowable uses of "or" and "while". However, when building the

```
DO1
the objective.n of this section.n is.v to safeguard.v occupants.n from illness.n
the objective.n of this section.n is.v to safeguard.v occupants.n from injury.n
[while] evacuating.g in an emergency.n

objective --> objective_phrase, phrase .
objective_id --> objective_token .
objective_phrase --> [the], [objective], [of], [this], [section], [is], [to] .

phrase --> verb, noun_phrase, prep, noun_phrase .
phrase --> verb, prep, noun_phrase .
phrase --> phrase, conj, phrase .

noun_phrase --> det, noun .
noun_phrase --> noun .

%     dictionary entries
conj --> [or] .
conj --> [while] .
det  --> [an] .
noun --> [emergency] .
noun --> [injury] .
noun --> [illness] .
noun --> [occupants] .
prep --> [from] .
prep --> [in] .
verb --> [evacuating] .
verb --> [safeguard] .
```

Manual Entry

BCA

Figure 5. Input Sentences and Resultant Grammar

structure of Objective D01(b) for example, the phrase starting with "while" is dependent on the verb while the "or" indicates disjunction (inclusive or). Consequently, the grammar rules have to explicitly distinguish between the scope of these conjunctions.
The second method under test is to use a finite state machine method which chooses the next step of the parsing process depending on the branches leaving the current node. This distinguishes between the two uses of "safeguard" in Objective D01(b) (Figure 6).
The design of the internal representation is also still in its initial stages. Currently, Objectives, Functional Requirements and Performance Requirements are stored as "facts":

```
objective((d:1:b), and(
                safeguard(occupants, illness),
                safeguard(occupants, injury, evacuating(emergency))
                ).
```

Notice that two different forms of the verb "safeguard" are used here, one taking two arguments and the other taking three arguments. The sense (or meaning) of safeguard is identical for both uses. This ability of verbs to take different numbers of arguments has also been well studied (ie Sowa, 1984).
Deemed-to-Satisfy Requirements are stored as rules:

```
Rule d:1.2:b:i
      If ((building.class = 2) ∨ (building.class = 3) …) ∧
           (building.effective_height > 25m))
      then (forall building.storey.num_of_fire_exit >= 2)
           and objective(d:1:b).satisfied = true .
```

This is all conveniently expressible in first order logic, which reduces computational complexity.

7 FURTHER WORK

While the amount of work that has been completed so far is not trivial, this work is still in the initial stages. As has been demonstrated in over 30 years of research effort, the interpretation of natural language text is a complex enterprise that requires a steep investment of effort before any significant return is achieved (Wilks et al, 1996).
Much work still needs to be done on identifying all of the information that must be stored in the dictionary to allow full interpretation of the BCA. This work will be readily applicable to other codes and standards.
The most appropriate method used to encode the grammar is still under investigation. However, the usefulness of the grammar itself in the interpretation of other codes and standards will be limited due to the differences in clause structure across different documents. The underlying method for generating the grammar will have wider applicability. Whether the effort justifies the return is a different matter.

8 REFERENCES

ABCB (1996) Building Code of Australia, Australian Building Codes Board

CSIRO (1999) BCAider Home Page,
http://www.dbce.csiro.au/ind-serv/brochures/bcaider/bcaider.htm

Fellbaum, C. (ed) (1998) WordNet: An Electronic Lexical Database, The MIT Press

Fenves, S., Wright, R. N., Stahl, F. I., and Reed, K. A. (1987). Introduction to SASE: Standard analysis, synthesis and expression. Technical Report NBSIR 872513, National Bureau of Standards, Washington, D.C.

Garrett, J. J. and Hakim, M. M. (1992). An object-oriented model of engineering design standards. *Journal of Computing in Civil Engineering*, 6(3):323–347

Rosenman, M. A. and Gero, J. S. (1985). Design codes as expert systems, Computer-Aided Design, 17(9):399-409

Russell, S. & Norvig, P. (1995). Artificial Intelligence: A Modern Approach, Prentice Hall

Sowa, J.F. (1984) Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley

Turk, Z. and Janez, D. (1995) Slovenia and computer representations of design standards and building codes, *International Journal of Construction Information Technology*, 3(1):55-71, 1995

Vanier, D. J. (1995) Canada and computer representations of design standards and building codes. *International Journal of Construction Information Technology*, 3(1):1-12, 1995.

Wilks, Y., Slator, B. & Guthrie, L. (1996) Electric Words: Dictionaries, Computers and Meanings, The MIT Press

Yabuki, N. and Law, K. H. (1993). An object-logic model for the representation and processing of design standards. *Engineering with Computers*, 9:133–159.