International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:6, 2008

MAYA SEMANTIC TECHNIQUE: A Mathematical Technique Used to Determine Partial Semantics for Declarative Sentences

Marcia T. Mitchell

Abstract—This research uses computational linguistics, an area of study that employs a computer to process natural language, and aims at discerning the patterns that exist in declarative sentences used in technical texts. The approach is mathematical, and the focus is on instructional texts found on web pages. The technique developed by the author and named the MAYA Semantic Technique is used here and organized into four stages. In the first stage, the parts of speech in each sentence are identified. In the second stage, the subject of the sentence is determined. In the third stage, MAYA performs a frequency analysis on the remaining words to determine the verb and its object. In the fourth stage, MAYA does statistical analysis to determine the content of the web page. The advantage of the MAYA Semantic Technique lies in its use of mathematical principles to represent grammatical operations which assist processing and accuracy if performed on unambiguous text. The MAYA Semantic Technique is part of a proposed architecture for an entire web-based intelligent tutoring system. On a sample set of sentences, partial semantics derived using the MAYA Semantic Technique were approximately 80% accurate. The system currently processes technical text in one domain, namely C++ programming. In this domain all the keywords and programming concepts are known and understood.

***Keywords*—**Natural language understanding, computational linguistics, knowledge representation, linguistic theories.

I. Introduction

THE purpose of this paper is to show that mathematical techniques can be used with natural language processing (NLP) to analyze and comprehend the content of web pages. The comprehension of web pages is useful for a variety of purposes, such as summarizing the results of web searches and information retrieval. The application presented here is part of a web-based, intelligent tutoring system (ITS). The system during its use acquires new knowledge, which further assists the student in various learning tasks.

The current research is an outline and model of an ITS that will have the capacity to run in the World Wide Web environment. If the content of the page is not in the knowledge base of the web-based tutorial program, then after the page has been examined thoroughly for relevance by the web-based tutorial program, the web-based tutorial program automatically adds it to that knowledge base. It is important to note that a World Wide Web tutor cannot be restricted to a closed world knowledge domain since students are always free to browse any page on the World Wide Web regardless of its relation to the knowledge domain of the tutor. If the page is totally irrelevant, its content should be ignored by the tutor. If there is relevant content already in the knowledge base of the tutor its representation may or may not be added either as part of the knowledge base or by reference. If there is relevant content not already in the knowledge base, it should be added to the knowledge base [13].

One challenge to developing a useful ITS is topical relevancy. If the content of the page is not in the knowledge base of the tutor, then the system automatically adds it to that knowledge base after the page has been thoroughly examined for relevance by the appropriate software components within the system. Relevance is determined based on the subject the student is learning. The system could use this knowledge for instruction and explanation purposes for students. The system's current knowledge domain is C++ programming; and its components, developed for the current research, can classify/recognize C++ keywords and programming concepts that are presented on web pages.

Current views on knowledge representation and semantic interpretation are also useful here, as theories of knowledge representation and semantic processing are an integral part of understanding how to develop new techniques for natural language processing.

II. Knowledge Representation

In connection with computer applications, [26] define knowledge representation as "the study of how to put knowledge into a form that a computer can reason with" [26]. Essentially, what is "knowledge" then? The term "ontology" is part of our answer: *The Oxford English Dictionary* (1997) defines ontology as "The science or study of being; the department of metaphysics which relates to the being or essence of things, or to being in the abstract." Reference [26] generally concurs, defining ontology as "a particular theory of the nature of being or existence" [26]. In the world, however, as [20] explain, knowledge about the world is permanently stored in the *ontology* [20]. In the author's program, the system's *ontology* is the collection of C++ statements that exist in the computer's knowledge base. Thus, in this author's ITS architecture, the ontology for the system for natural

Manuscript received March 13, 2005.
M. T. Mitchell is with Saint Peter's College, Department of Computer and Information Science (phone: 201-915-4960; e-mail: mmitchell@spc.edu).

language processing is defined in a file that contains C++ statements and programming concepts about language along with their associated extended ASCII character (see Figure 1). The character may be mathematical, graphic or foreign.

The most commonly used techniques for representation are frames, semantic networks and logical forms. Reference [2] indicates that a frame is a "cluster of facts and objects that describe some typical object or situation, together with specific inference strategies for reasoning about the situation". However, frame-based representation is appropriate when the domain has many possible interpretations. One disadvantage of frames, "…allowing alterations of slots [placeholders] without control [proved to be…] one of the major problems in the frame schema as the properties of the frame that the object inherits …can be cancelled" [11]. This author rejected the use of frames due to this disadvantage. This author's system limits itself to language consisting of simple commands and unambiguous statements, thus circumventing the use of frame-based representations to interpret simple technical texts. C++ texts do not permit multiple interpretations.

According to [2], a semantic network is a "graph with labeled links between labeled nodes. The nodes represent word senses [meanings], or abstract classes of senses, and the links represent semantic relationships between the senses" [2]. According to [26], both frames and semantic networks use "objects" as "nodes in a graph, that these nodes are organized in a taxonomic structure, and that links between nodes represent binary relations. In frame systems the binary relations are thought of as slots in one frame that are filled by another, whereas in semantic networks, they are thought of as arrows between nodes" [26]. These arrows facilitate the directional flow of information. According to [11], semantic nets are used to represent declarative knowledge. Semantic nets were developed to model both human memory and understanding of language [11]. However, "[s]emantic nets have limitations in representing knowledge when it comes to definitive knowledge…." [11]. One disadvantage according to [11], is the inheritance problem as the node properties can be changed. Nodes and links must be strictly defined to overcome this disadvantage [11].

Logical forms are abstract representations of the semantic structure of a sentence [19]. Reference [2] states that logical forms are used for "represent[ing]…the context-independent meaning of a sentence. The logical form encodes possible word senses and identifies the semantic relationships between the words and the phrases [2]. Reference [2] states that, the language of logical forms or functions resemble first-order logic [2]. According to [17], a declarative sentence is "...one that belongs, but virtue of its grammatical structure, to the class of sentences whose members are used, characteristically, to make statements...." For the purposes of this discussion, declarative statements are reduced to their indivisible parts--subject (noun), verb, and object and a mathematical technique is then implemented. Indeed, according to [23], a sentence is understood by a "process of finding the subjects, verbs, objects, and so on, [and] that [process] takes place unconsciously" [23].

MAYA, however, does not use logical forms or functions for knowledge representation. It takes another approach. By using extended ASCII characters such as mathematical, graphical or symbol, the MAYA Semantic Technique represents knowledge. Extended ASCII characters use one byte to represent a C++ keyword or a programming concept for the C++ programming language in order to be abstracted and computed. The system uses extended ASCII characters to represent knowledge, which, in turn, facilitates the mathematical/statistical processing of text.

III. Traditional Methods Used in Natural Language Processing

According to [27], semantic interpretation is the process of determining the meaning of the input [in this case, words, phrases, and sentences]. Reference [27] explains several methods of determining semantics by using situational semantics, procedural semantics, logical form, or Montague's techniques on semantic interpretation. Richard Montague's semantic system uses a one-to-one correspondence between a set of syntactic rules and a set of semantic rules: that is, associating the meaning of the sentence with the syntax of the sentence. "Montague employed a kind of extended categorical grammar and a syntax-semantics correspondence in which function-argument application play a central role" [21]; in short, where nouns, verbs, adjectives, prepositions, and the other parts of speech are grammatical categories. However, [27] states that, Montague's semantic is not possible to implement because direct implementation is computationally impractical at this time [27]. The first reason Montague's semantics is not possible to implement is that it throws around huge sets, infinite objects, functions of functions, and piles of possible worlds with great abandon. The second reason is that truth-conditional semantics is inadequate for Artificial Intelligence (AI) [27]. Finally, Montague's semantics is not used with C++ programming because, as noted above, doing so is computationally intractable.

The artificial language of C++ assesses the truth or falsity of a statement in conditional statements; but the truth or falsity of the natural language statement is of no particular concern to the system. C++ does not "care" if the natural language statement is a lie. Indeed, truth-conditional semantics is defined by [16], according to [5], as giving "…an account of the meaning of a sentence [in order to]…specify the conditions under which it would be true or false ... [about that which]…it purports to describe" [5].

Situational semantics--the attempt to formalize the idea of a situation in the real world as a suitable semantic object [27]--is examined by [1]. "Some activities pertaining to language include talking, listening, reading, and writing. What is common to these…activities is that they convey information. These activities are *situated;* they occur *in* situations [occurrences, events, happenings] and they are *about* situations. When uttered at different times by different speakers, a statement can convey different information to a hearer and hence can have different meaning" [1]. C++

programming does not include such situations because the information in C++ programming statements admits only one interpretation.

Finally, [27] describes procedural semantics as production rules that "translate the parsed input into procedure calls that operate upon a database, and the meaning of a sentence is identified with the corresponding procedure call" [27]. For the developer, there are problems with production rules in procedural semantics because production rules are *ad hoc* and non-computational. The rules often look for specific words as their trigger. If the input word is changed, then the rules may not work [27]. Another problem with symbolic processing techniques is that it cannot be used to processing unrestricted text. This research does not use production rules because production rules do not execute consistently.

IV. Syntactic Structure

According to [12], "The meaning of a sentence is not based solely on the words that make it up, it is based on the ordering, grouping, and relationships among the words in the sentence" [12] Reference [2] says that syntactic structure refers to the way the words in a sentence are related to one another [2]. Thus, the syntactic structure of a sentence is derived by parsing the sentence. Parsing is the process of identifying the sentence's grammatical structures, such as subject, predicate and modifiers. These large structures are more specifically broken down into "parts of speech;" and in English there are eight: nouns, pronouns, verbs, prepositions, articles, adjectives, adverbs, and conjunctions. According to [22] "a part of speech, then, is not a kind of meaning; it is a kind of token that obeys certain formal rules, like a chess piece or a poker chip. A noun, for example, is simply a word that does nouny things…" [22]

Another method of natural language processing is described by [18]. They describe statistical methods that can be applied to a corpus. Some of the topics covered include part-of-speech tagging, probabilistic context-free grammars, and probabilistic parsing. MAYA uses statistical methods for semantics interpretation.

The techniques described in this study also extend the concepts described by [18] to include the derivation of partial semantics using statistical methods. This study will demonstrate how mathematical techniques are used in a natural language processing program to determine the contents of a web page and to derive partial semantics from that content. The mathematical techniques used categorize the vocabulary in order to derive partial semantics and summarize the content. Both traditional and statistical methods rely on syntactical structure.

V. Linguistics Theories

The attempt to convert rules that describe the patterns in language into mathematical terms has so far proven to be unsuccessful when considering language as a whole. However, several linguists believe that mathematics hold promise as it can be used to analyze the patterns and relations between categories and phrases in a sentence. Frege showed that certain structures can be described mathematically and logically. This raises the prospect that restricted areas of language may be amenable to the algebraic treatment suggested by de Saussure, Louis Hjelmslev, and Chomsky [6]. Hjelmslev was concerned with "establishing a set of formal definitions from which theorems can be derived for the purpose of [computationally] describing the patterns of languages" [8]. According to [8], Hjelmslev thought the description of language should be scientific "in terms of relations between units, irrespective of any properties which may be displayed by these units but which are not relevant [or deductible] to the relations [among linguistic units]…" [8] Hjelmslev thought "…if we can isolate the relevant relations among linguistic units, we should have as powerful a theory of language as we have in mathematics" [8].

This paper describes an attempt to use computational techniques in a restricted area, as applied to declarative sentences used in an ITS context.

A. Statistical Linguistics

Statistical linguistics is broadly classified as a branch of mathematical linguistics. Mathematical linguistics is concerned with applying formal and mathematical properties to language [10] and is comprised of two distinct areas of research. One area investigates the statistical structure of texts, and the other area investigates the construction of mathematical models of phonological and grammatical structures in language (referred to as statistical and algebraic linguistics, respectively.) In fact, according to [6], de Saussure believed that language can be conceived of as a type of algebra, that is, an expression of relations between grammatical categories. In other words such relations can be expressed by algebraic formulas, proportions, and equations [6]. The research presented here demonstrates grammatical relationships in mathematical terms; however, these relationships are limited to those that occur in declarative sentences in technical texts.

B. Rational and Empirical Approaches to Natural Language Processing

There are two approaches to natural language processing that are currently used: rationalism and empiricism. Rationalists believe that "… a significant part of the knowledge in the human mind is not derived by the senses but is fixed in advance, presumably by genetic inheritance. …Within artificial intelligence, rationalist beliefs can be seen as supporting the attempt to create intelligent systems by handcoding into them a lot of starting knowledge and reasoning mechanisms, so as to duplicate what the human brain begins with" [18]. An empiricist approach to NLP suggests that one can learn the complicated and extensive structure of language by specifying an appropriate general language model. This model induces the values of parameters by applying statistical techniques, pattern recognition, and machine learning methods to a large amount of language use [18].

This author's research stresses the use of an empiricist approach to see if mathematical models can be used to

determine semantics and if the results of these operations can be implemented by a web-based ITS. The system developed by this author applies mathematical/statistical techniques to the extended ASCII character and performs pattern-matching techniques to determine the content of a web page. The system derives partial semantics from the extended ASCII character representation and performs concise summarizing. Hence, statistical analysis for natural language processing is suited for text as the text itself exists on the internet.

C. Statistical Linguistics

Statistical linguistics is appropriate to use with declarative sentences. Since technical texts rely on statements of fact, the structure of declarative sentences lends itself to statistical analysis and application of the MAYA Semantic Technique. This research demonstrates there are relationships between verbs and objects in the declarative sentences occurring in technical texts that can be expressed mathematically. This research uses computational linguistics, an area of study when a computer processes natural language and discerns the patterns that exist in simple declarative sentences that are most likely to be used in technical texts. Techniques in statistical linguistics also look at the word patterns and frequency of parts of speech. Usually, the pattern of subject, verb and object is common to most declarative sentences; thus this research has tried to develop statistical techniques that can be used to derive partial semantics.

VI. Application of Knowledge Representation Techniques Enabling Maya Semantic Processing

The domain is C++ programming, currently implemented at a high level in a file that lists C++ keywords and programming concepts in the form of an extended ASCII character dictionary. The dictionary contains extended ASCII characters, each representing a C++ keyword and/or a C++ programming concept. The domain is defined at a high level because once the C++ keywords and programming concepts have been converted to extend ASCII characters the natural language processing program processes the text quickly.

A sample of the file is shown in Figure 1.

η
auto
f
bool
β
break
γ
const
φ
case
μ
char
κ
cout
…

Fig. 1 Sample of the extended ASCII character dictionary file

The program reads each sentence and converts C++ keywords and programming concepts into a byte code. For example, the C++ statements "if else" are abstracted and viewed as "ÑÑ ÊÊÊÊ." Byte code representation creates an abstraction of the text.

The program also has converted its knowledge base into extended ASCII character symbols. Pattern-matching techniques are employed to determine if any programming code found on the web page is different from the existing templates in the knowledge base. For example, the actual programming code from the web page is processed as illustrated in Figure 2.

```
if (x < 0)
   sign = -1;
else
  if (x == 0)
     sign = 0;
else
      sign = 1;
```

Fig. 2 Actual programming code from a web page [14]

The process of pattern matching involves several steps. The first step is to remove the conditions and assignment statements, leaving the "if else statement" template:

```
if (   )
        ;
else
 if (   )
        ;
else
          ;
```

Fig. 3 Template of the "if else statement"

Then the program converts the template above to extended ASCII symbols and, as noted, uses pattern-matching techniques to determine whether the templates are the same as those of the web page. When using extended ASCII characters, the program does not have to process the different characters that make up a word. Thus, by abstracting the template into extended ASCII characters, the program can quickly and easily process the content of the web page. The code in Figure 3 has been converted to extended ASCII characters in Figure 4:

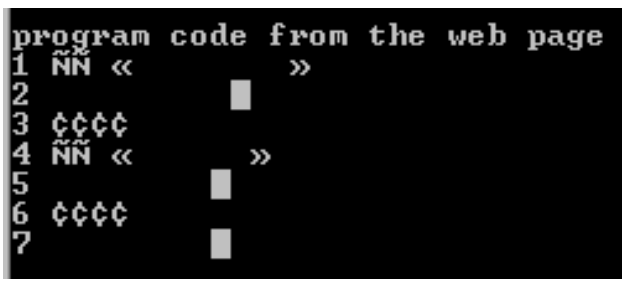

Fig. 4 The program's view of the templates

If the template is not found within the domain knowledge base, the template will be added after examination by the appropriate program within the natural language processing system. The templates in Figure 5 are part of the domain before any particular web page is read.

1. if ( ) { ; } else { ; }
   compound if then else statement
2. if ( ) ; else ;
   if then else statement
3. if ( ) ; else { ; }
   simple if and compound else statement
4. if ( ) { ; }
   compound if then
5. if ( ) ;
   if then statement

Fig. 5 Conditional statement templates

The "if else" selection structure.

The text below uses the if else selection structure as a possible web page for illustrative purposes. The if selection structure performs an indicated action only when the condition is true; otherwise the action is skipped. The if/else selection structure allows the programmer to specify that a different action is to be performed when the condition is true than when the condition is false [7].

Once the web page and knowledge base template are coded in extended ASCII format, comparisons using a pattern-matching technique are then made between the codes in the web page and those in the knowledge base to determine if they are equivalent.

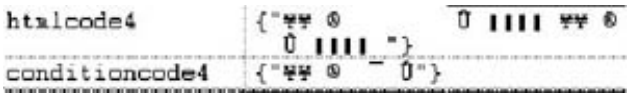

Fig. 6 Comparison of the domain and the web page templates

Figure 6 shows "htmlcode4," the name of a list, and it contains items in web page template. The "conditioncode4" list, which is the first template from the knowledge base, is compared to "htmlcode4."

The program view of the template "conditioncode4" is illustrated in Figure 7. This figure shows the knowledge base before the programming code has been added to its template.

Fig. 7 Template converted to extended ASCII character

Abstracting at a high level the program processes quickly because the technique only has to compare code rather than whole words or phrases.

Fig. 8 Shows the Web ITS view of the example text as a web page converted to extended ASCII characters.

VII. Components of the Natural Language Processing Program

This section introduces the programs that are involved in processing text from the web page. The one program that will not be discussed here is the HTML program since it extracts content rather than processes it. This paper focuses on MAYA[1]. The Part of Speech Program; the Knowledge Representation Program; and the author's MAYA Semantic Technique are all part of Natural Language Processing Programs.

A. Part of Speech Tagger Program

The Part-of-Speech Tagger program reads the file that contains the sentences extracted from the web page. This program determines the part of speech for each word in the input line. Currently, the natural language processing program does not check whether a source web page has incorrect syntax.

B. The Process-of-Tagging

The natural language processing program calls this program and sends the web page along so that the Part of Speech Tagger Program can process it. The processing of web pages is organized into several steps. The first step is to tag the text that comes from the web page. The part-of-speech tagging process is performed by the [4] [25] part-of-speech tagger (see [4] [25] webpage for further information re: the tagger.) The current system then sorts information into twelve categories, eight of which are parts of speech, two of which are technical terms which facilitate the program and the last category is the cardinal. The inclusion of categories specific to the tutor's subject, currently C++, helps in processing technical text. For C++, the new categories to be tagged are "program," "computer" and "cardinal." The "program" category is used to represent programming concepts, and the "computer" category is used to represent C++ keywords. This is done because the "program" category is a refinement of the original category. "Program," "computer," and "nouns" (in the general English grammar sense) are treated as key categories in the MAYA Semantic Technique.

Figures 9 and 10 respectively show some examples of words that are "program" and "computer" categories:

argument	assignment	condition
nested	operator	recursion
counter	repetition	iteration
loop	statement	variable

Fig. 10 "Program" category

auto	char	const	else
for	if	int	sizeof
struct	class	public	private

Fig. 10 "Computer" category

The system reads the web page file that contains the text, tags the contents of the web page file and saves the categories in another file. The system uses the following lexicon to determine the category for each word:

- adjective
- adverb
- article
- auxiliary verb noun
- cardinal
- computer (C++ keywords)
- conjunction
- noun
- preposition
- program (programming concepts)
- pronoun
- verb

The part-of-speech tagger determines the categories for each word in the sentence and saves the results as a file.

C. Example of the Output of the Part-of-Speech Tagger Program

The following text is an example of the part of speech tagging from the domain knowledge base [24].

"the if else statement lets a program decide which of two statements or blocks is executed"

The sentence tagged by the Parts-of-Speech Tagger Program is as follows:

*article computer computer program verb article noun verb pronoun preposition noun program conjunction noun verb verb

The asterisk indicates the beginning of the sentence that was derived from the Part-of-Speech Tagger Program. Currently, the Part-of-Speech Tagger Program does not differentiate relative pronouns from other types of pronouns.

Each category is converted to an ASCII character for easy processing. Hence, the sentence is converted to:

a c c g v a n v r p n g o n v v

The categories are designated as follows:

a=article	c=computer	d=adverb
i=infinitive	j= adjective	g=program
l=cardinal	n=noun	o=conjunction
p=preposition	r=pronoun	v=verb

Fig. 11 Categories for C++ text

D. Example of the Domain Knowledge Base taken from [24]

Input file to the natural language program

"the if statement lets a program decide whether a particular statement or block is executed"
"the if else statement lets a program decide which of two statements or blocks is executed"
"it is an invaluable statement for creating alternative courses of action"
"if the test condition is true or nonzero the program executes statement one and skips over statement two"
"otherwise when test condition is false or zero the program skips statement one and executes statement two instead"

Output from [4][25] Part-of-Speech Tagger

*the/DT if/IN statement/NN lets/VBZ a/DT program/NN decide/VB whether/IN a/DT particular/JJ statement/NN or/CC block/NN is/VBZ executed/VBN

*the/DT if/IN else/JJ statement/NN lets/VBZ a/DT program/NN decide/VB which/WDT of/IN two/CD statements/NNS or/CC blocks/NNS is/VBZ executed/VBN

*it/PRP is/VBZ an/DT invaluable/JJ statement/NN for/IN creating/VBG alternative/NN courses/NNS of/IN action/NN

*if/IN the/DT test/NN condition/NN is/VBZ true/JJ or/CC nonzero/VBG the/DT program/NN executes/VBZ statement/NN one/CD and/CC skips/VBZ over/IN statement/NN two/CD

*otherwise/RB when/WRB test/NN condition/NN is/VBZ false/JJ or/CC zero/CD the/DT program/NN skips/VBZ statement/NN one/CD and/CC executes/VBZ statement/NN two/CD instead/RB

Output file from the natural language program

*article computer program verb article noun verb preposition article adjective program conjunction program verb verb
*article computer computer program verb article noun verb pronoun preposition cardinal program conjunction noun verb verb
*pronoun verb article adjective program preposition verb noun noun preposition noun
*computer article program program verb program conjunction program article noun verb program cardinal conjunction verb preposition program cardinal
*adverb adverb program program verb program conjunction cardinal article noun verb program cardinal conjunction verb program cardinal adverb

E. Example of Analysis of Partial Semantics

The natural language processing program reads the file that contains the categories and converts each category into extended ASCII character. The natural language processing program then determines the subject of the sentence.

The first sentence in the paragraph reads

"the if statement lets a program decide whether a particular statement or block is executed"

The natural language program determines the categories for this sentence as follows:

*article **computer program** verb article noun verb conjunction article adjective program conjunction program verb verb

The natural language processing program then uses the MAYA Semantic Technique to perform partial semantics.

VIII. Implementation of the MAYA Semantic Technique

For the purpose of this article, the only module implemented in the domain knowledge base is conditional statements such as those associated with "if" and "if-then."

Once the sentence has been broken down into categories, the MAYA Semantic Technique is applied which involves the following four steps: Step one determines the subject of the sentence; step two performs frequency analysis on the categories and applies a reduction technique; step three determines which category, computer (C++ keywords) or program (programming concepts), or noun has the highest frequency. The technique identifies "key" categories as more important than others. The key category with the highest frequency identifies the object of the sentence. Step four, the final step, determines the main verb of the sentence. The result is a web page converted to extended ASCII character symbols. Figure12 text below is used as an example web page for illustrative purposes [7].

```
<html>
<head><title>Sample Web Page </title></head>
<body>
<pre>
if else statement
The if selection structure performs an indicated action only
when the condition is  true; otherwise the action is skipped.
The if/else selection structure allows the  programmer to
specify that a different action is to be performed when the
condition is true than when the condition is false
if ( hours  < 25 )
   rate =25;
else if ( hours  < 50 )
  rate = 50;
else if ( hours <  75 )
  rate = 75;
else
  rate = 100;
</pre>
</body>
</html>
```

Fig. 12 Example of a web page

The natural language processing program must now determine if the web page is relevant to what the student is studying by comparing the content of the web page with the existing content of the domain knowledge base.

It takes the appropriate portions of the domain knowledge base and abstracts it into extended ASCII characters. The knowledge base extended ASCII character file and the web page extended ASCII character file are compared to see if there is a match. Matching involves first determining the frequencies of the C++ words and the programming concepts, then comparing these frequencies to ascertain whether the files are similar.

Since both the web page and the domain knowledge base module contain material on condition statements, the natural language processing program concludes that the two files are similar.

Figure 13 shows the result of this comparison.

Fig. 13 Results after comparison of web page and knowledge base

A. Step One of the MAYA Semantic Technique

The MAYA Semantic Technique begins by grouping related categories. Hence, adjectives are grouped with nouns, and adverbs are grouped with verbs. The natural language processing program then determines the subject of the sentence. The natural language processing program includes functions to help to accurately determine the subject of the sentence. Once the subject of the sentence has been identified, the subject of the sentence is put on the semantic list for either the knowledge base or the web page. The rest of the sentence is then processed to determine the verb and the object/complement.

The MAYA Semantic Technique uses the following technique to determine the sentence subject. The MAYA Semantic Technique locates the first verb in the sentence. Next, MAYA Semantic Technique tries to determine whether any key categories ("computer," "program" or "noun") precede the first verb.

There are several sub-steps for determining the subject of the sentence. MAYA Semantic Technique may be applied as follows:

1) Sub-step #1

*article **computer program** verb article noun verb
conjunction article adjective
program conjunction program verb verb

Since declarative sentences follow the subject-verb-object pattern, the natural language processing program determines

how many key categories there are, along with their locations in the sentence.

2) Sub-step #2

The next step is to group adjacent categories that are alike.
*article **program** verb article noun verb conjunction article adjective program conjunction program verb
For example, "**program**" is the category that is the subject of the sentence.

Subject = **program** = if statement

The categories that represent the subject of the sentence are then removed, resulting in:

*article **program** verb article noun verb conjunction article adjective
program conjunction program verb

* verb article noun verb conjunction article adjective
program conjunction program verb

3) Sub-step #3
The frequency of each category is shown below as an exponent where S = sentence:

$S = article^1 + program^2 + verb^3 + noun^1 + conjunction^2 + adjective^1$

B. Step Two of MAYA Semantic Technique

Before the reduction technique is applied, the natural language processing program needs to determine which key category remaining in the sentence has the highest frequency. The natural language processing program assigns priority levels to the three key categories. The "computer" key category has the highest priority because it reflects C++ keywords; next is the "program" category because it reflects programming concepts, and finally, the noun. If two key categories have equally high frequency, then the natural language processing program determines which key category has the higher priority. Higher priority is given to the category which includes the type of words the text is about. The categories with the highest frequencies become the object of the sentence. If "program" has the highest frequency, then the reduction technique is applied based on the category "program."

The sentence with all the frequencies is S:

$S = article^1 + program^2 + verb^3 + noun^1 + conjunction^2 + adjective^1$

C. Step Three of the MAYA Semantic Technique

The result of this first reduction is to determine the verb and object of sentence S_1:

$S_1 = program^1 + verb^2 + conjunction^2$

S_1 cannot be reduced further because the category "program" is in its lowest terms. This reduction is Step 3 since the category is in its lowest terms.

The main verb is the second verb V_1 in this sentence S_1. The main verb is determined after the system locates the sentence's object. The frequency for each category is now reduced by 1. S_1 cannot be reduced further because the category "program," the only remaining noun, is in its lowest terms, so it becomes the object.

Therefore in the sentence, "the if statement lets a program decide whether a particular statement or block is executed," the following is found:

Subject = computer program = if statement
Verb = $verb^2$ = decide ← Second **verb** in the

Since there is a conjunction, the object must include the second "program" category, which is the word "block."

Object and/or Complement = $program^1$
= statement or block
First and second program in the sentence

In determining the partial semantics the complementary information in the sentences are ignored. This information does not help in determining relevancy because relevancy is defined by the system's domain specific lexical knowledge (the ontology).

The second sentence in the paragraph reads:
"the if else statement lets a program decide which of two statements or blocks is executed"

*article **computer computer program** verb article noun verb pronoun preposition cardinal program conjunction program verb verb

The next step is to group adjacent categories that are alike.

*article **program** verb article noun verb pronoun preposition cardinal program conjunction program verb

Subject = **program** = if else statement
* verb article noun verb pronoun preposition cardinal program conjunction program verb

$S = article^1 + program^2 + verb^3 + noun^1 + pronoun^1 + preposition^1 + conjunction^1 + cardinal^1$
$S_1 = program^1 + verb^2$

Subject = computer computer program = if else statement
Verb = $verb^2$ = decide
Object and/or Complement = $program^1$ = statements or block

The third sentence in the paragraph reads:

"it is an invaluable statement for creating alternative courses of action"

*pronoun verb article adjective program preposition verb noun noun preposition noun

The next step is to group adjacent categories that are alike.

*pronoun verb article program preposition verb noun preposition noun
Subject = **computer computer program**

The pronoun refers back to a previous expression. The word "it" will be replaced with the words "if else statement."

$S = verb^2 + article^1 + program^1 + preposition^1 + noun^2$
$S_1 = verb^1 + noun^1$

Subject = computer computer program = if else statement
Verb = $verb^2$ = is
Object and/or Complement = $program^1$ = invaluable statement

The fourth sentence in the paragraph reads:
"if the test condition is true or nonzero the program executes statement one and skips over statement two"

conjunction article **program program** verb program conjunction program article noun verb program cardinal conjunction verb preposition program cardinal

The next step is to group adjacent categories that are alike.
conjunction article **program** verb program conjunction program article noun verb program conjunction verb preposition program

Subject = **program** = test condition

conjunction article **program** verb program conjunction program article noun verb program conjunction verb preposition program

verb program conjunction program article noun verb program conjunction verb preposition program

$S = verb^3 + program^4 + article^1 + noun^1 + preposition^1 + conjunction^2$
$S_1 = verb^2 + program^3 + conjunction^1$
$S_2 = verb^1 + program^2$
$S_3 = program^1$

The verb is taken from S_2 which is the first verb in the sentence.

The object is taken from S_3 which is the first program (category) in the sentence.

Subject = program program = test condition
Verb = $verb^1$ = is
Object and/or Complement = $program^1$ = true or nonzero

The fifth sentence in the paragraph reads:
"otherwise when test condition is false or zero the program skips statement one and executes statement two instead"

*adverb adverb **program program** verb program conjunction cardinal article noun verb program cardinal conjunction verb program cardinal adverb

The next step is to group adjacent categories that are alike.

*adverb **program** verb program conjunction cardinal article noun verb program conjunction verb program adverb

Subject = **program**= test condition

* verb program conjunction cardinal article noun verb program conjunction verb program adverb

$S = adverb^1 + conjunction^2 + verb^3 + program^3 + cardinal^{1+} article^1 + noun^1$
$S_1 = verb^2 + program^2$
$S_2 = verb^1 + program^1$

Subject = program = test condition
Verb = $verb^1$ = is
Object and/or Complement = $program^1$ = false or zero

The natural language processing program displays the partial semantics for the sentences from the web page:
if statement decide statement or block
if else statement decide statements or blocks
if else statement is invaluable statement
test condition is true
test condition is false

D. Step 4 of the MAYA Semantic Technique

The result of the partial semantics is placed in a list. The list is used by the natural language processing program in order to compare the results of the partial semantics from both the knowledge base file and the web page file in order to determine if the files are equivalent. By taking the subject of each sentence in the knowledge base file and matching it to the subject in the web page file comparisons are made.

Natural language processing includes a set of routines to analyze "sentences," in the sense of strings of data. For this project, we also want to analyze natural language in order to inform students who are engaged in learning C++ programming via an ITS. Thus, a well-conceived natural language processing program could perform two functions: The first function reads and analyzes web pages, and the

second function facilitates communication with students, a benefit of performing the first function.

IX. Manual Examination Verifying the MAYA Semantic Technique

Two different subject matters, biology and physics, are used here to suggest that the MAYA Semantic Technique is not C++ dependent. The sentences below were not implemented in the computer; they were analyzed by inspection and are included to demonstrate that the MAYA Semantic Technique can possibly work in other scientific domains. Further research will be required to support this suggestion across other subject domains.

A. Biology

In the example below, the inclusion of one new category helps in processing biology text. This new category is a specialized noun, a biological term called "bio," which is represented as "b" which is one byte. "Bio" stands for any noun and/or its attributes that are specific to biological terminology in the text. The following example is a sentence from the [3] biology text:

Example 1

The spacing along the helix axis from one base pair to the next is 2.9Å.

The partial semantics of the sentence without removing the prepositional phrases follows:

*article **noun** preposition article bio bio preposition cardinal adjective noun preposition article noun verb bio

The next step is to group adjacent categories that are alike.
*article **noun** preposition article bio preposition cardinal noun preposition article noun verb bio

Subject = **noun** = spacing

* preposition article bio preposition cardinal noun preposition article noun verb bio

S_1 = preposition^3 + article^2 + bio^2 + cardinal^1 + noun^2 + verb^1

S_2 = preposition^2 + article^1 + bio^1 + noun^1

Subject = noun = spacing
Verb = verb^1 = is
Object and/or Complement = bio^1 = helix axis

Obviously, the partial semantics above are incorrect. There are several prepositional phrases that separate the subject from its verb; thus, the partial semantics does not represent the sentence. Prepositional phrases are properties or attributes of the subject of the sentence.

*article prepositionalphrase prepositionalphrase prepositionalphrase verb bio

Therefore, prepositional phrases are removed before the first verb as in the following example:

* verb bio

S_1 = bio^1 + verb^1

Subject = noun = spacing
Verb = verb^1 = is
Object and/or Complement = bio^1 = 2.9Å

In order to derive partial semantics using the MAYA Semantic Technique, all the prepositional phrases that occur before the first verb in the sentence must be identified and removed.

Here are two more examples from the [3] biology text:

Example 2

DNA is a linear molecule.

bio verb article adjective bio

verb article adjective bio

S= verb^1 + article^1 + adjective^1 + bio^1

Subject = bio = DNA

Verb = verb^1 = is
Object and/or Complement = bio^1= linear molecule

Example 3

Furthermore, a new structural form of DNA called Z-DNA has been discovered.

adverb article adjective adjective noun preposition bio participle bio auxiliary verb verb

The next step is to group adjacent categories that are alike.

adverb article **noun** preposition verb

Subject = noun = new structural form

verb

S_1= verb^1

Subject = form
Verb = has been discovered

B. Physics

A different subject matter, physics, is used next to suggest, again, that the MAYA Semantic Technique is not C++ dependent. The inclusion of one new category helps in processing this text. This new category called "physicterm" contains specialized nouns and their physical properties.

The text below is from [9]. This sentence demonstrates the MAYA Semantic Technique. The sentence contains prepositional phrases after the main verb of the sentence.

By the principle of energy we must therefore have
$E\gamma^{h}/c^{2} = M'\gamma h - M\gamma h$, or $M' - M = E/c^{2}$

preposition article noun preposition noun pronoun verb adverb verb physicterm conjunction physicterm
preposition article noun preposition noun pronoun verb physicterm conjunction physicterm
prepositional phrase prepositional phrase pronoun verb physicterm conjunction physicterm

Subject = pronoun = we
verb
physicterm conjunction physicterm

$$S_1 = \text{verb}^1 + \text{physicterm}^2 + \text{conjunction}^1$$

$S_2 = \text{physicterm}^1$

Subject = we
Verb = must therefore have
Object and/or Complement = $E\gamma^{h}/c^{2} = M'\gamma h - M\gamma h$ or $M' - M = E/c^{2}$

The results of the sentences on the previous pages analyzed by the MAYA Semantic Technique are shown in Figure 14.

Fig. 14 The MAYA Semantic Technique has been tested with pages from *[3] and ** [9]

The MAYA Semantic Technique does not require extensive processing to determine partial semantics for C++ technical text. Two new categories, "program" and "computer," have been presented in order to determine partial semantics. In the final step, the natural language processing program abstracts the text into extended ASCII characters, and then performs mathematical/statistical techniques. Based on these examples, the MAYA Semantic Technique can derive partial semantics and can be used for knowledge acquisition on the World Wide Web with technical texts.

The MAYA Semantic Technique was also applied to text from pages in biology and in physics books.

X. Computational Aspects of the MAYA Semantic Technique

C++ programming uses discrete data. In C++ programming, one can simply count the number of occurrences of each category. The program uses frequency distribution by grouping categories. After the frequency distribution takes place, reduction techniques are implemented to interpret the partial semantics.

S is the sentence that contains categories such as:

a=article	c=computer	d=adverb
g=program	i=infinitive	j= adjective
l=cardinal	n=noun	o=conjunction
p=preposition	r=pronoun	v=verb

Fig. 15 Categories for C++ text

The choice of letters for the C++ programming categories is arbitrary because the letters must be unique and can not be repeated.

The superscript represents frequency count variables, currently arbitrarily implemented as value from 0 to 20.

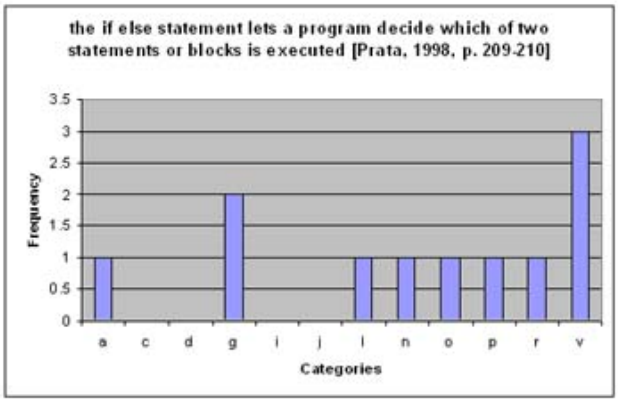

Fig. 16 Frequency distribution of a sentence

S = $\text{article}^2 + \text{program}^2 + \text{verb}^3 + \text{noun} + \text{pronoun} + \text{preposition} + \text{conjunction} + \text{auxiliary} + \text{adjective}$

Source	Number of Sentences	Number correct	Percent of partial semantics that was correct
*Introduction to Protein Structure	23	22	95
**The Principle of Relativity	14	13	92

where "S" is the sentence.

The MAYA Semantic Technique derives partial semantics for sentences using the following steps:

1.) It determines the subject of the sentence.

2.) It determines the frequency of the categories remaining.
3.) It determines which key categories ("computer," "program" or "noun") have the greatest
frequency count within the sentence. (The count is written as superscript over the categories, as indicated in examples.)
4.) It reduces each category by one with each iteration until the key category is in its lowest term
(i.e., until the count is one). The categories are reduced by subtracting one from each exponent; and
5.) It examines the count of the verb after all reductions. This calculation indicates which verb in the sentence is to be used as the main verb (e.g., v^2 means the second verb of the sentence.)

The reduction technique is as follows

$$S = a^{s} + c^{e} + d^{q} + g^{f} + i^{h} + j^{m} + l^{kk} + n^{k} + o^{y} + p^{b} + r^{w}$$

If $\{n_1, p_9, c_{10}\} = 1$ then stop, else $\{n_k = n_{k-1}, p_b = p_{b-1}, c_e = c_{e-1}\}$

Subtract one from each count until the count of the key category is one.

$$S_1 = a^{s-1} + c^{e-1} + d^{q-1} + g^{f-1} + i^{h-1} + j^{m-1} + l^{kk-1} + n^{k-}$$

a=article **c=1**	c=computer **key c=2**	d=adverb **c=3**
g=program **key c=4**	i=infinitive **c=5**	j= adjective **c=6**
l=cardinal **c=7**	n=noun **key index c=8**	o=conjunction **c=9**
p=preposition **c=10**	r=pronoun **c=11**	v=verb **c=12**

Fig. 17 Categories for C++ text

Been a new Example of a sentence being reduced:
S = conjunction[1] + verb[3] + program[4] + article[2] + noun[3] + preposition[1]+ conjunction[2]
S_1 = verb[2] + program[3] +article[1] + noun[2] + conjunction[1]
S_2 = verb[1] + program[2] + noun[1]
S_3 = program[1]

S is the structure of the sentence under investigation where a, c, d, g, i, j, l, n, o, p, r, and v represent the categories making up S, and s, e, q, f, h, m, kk, k, y, b, w, and z represent the count of each category of S. S_1 is the first reduction of the sentence S derived by reducing each count of each category by one. The letter *n* represents the highest count of the key ("computer," "program," or "noun") category. Therefore, S_n is the nth and final reduction of sentence S.

If computer is the key category then:
verb = { v | max c_2 }

XI. Evaluation of the Maya Semantic Technique

The C++ language contains declarative statements and states propositions that have precise factual meanings. Further, syntactic structure contains a pattern that can be discerned. Therefore, it is possible to apply statistical/mathematical techniques to "sentences" in C++ language in order to derive partial semantics.

In summary, the reason the MAYA Semantic Technique works for technical text is that the syntactic structure of C++ text is unambiguous, and only a limited number of parts of speech are contained in C++ text.

A. The Structure of Declarative/Proposition Statements

The MAYA Semantic Technique also works because declarative statements have an identifiable structure. According to [15], a "logical form of a thing depends upon its structure, or the way it is put together; that is to say, upon the way its several parts are related to each other [15]. Reference [15] states that nouns are represented as elements and verbs are represented as relations in a proposition [15]. "The elements which are connected by a relation are called, its terms" [15]. Elements are used to describe terms. These elements are properties or attributes of the terms. Reference [15] states that "the simplest logical structures are those expressed by propositions that mention just one relation and its terms" [15].

The MAYA Semantic Technique derives two terms and one relation term for each sentence. The first term is the subject of the sentence, and the second term is the object. The relation term is the verb in the sentence. The reason the MAYA Semantic Technique reduces each frequency count by one is that there is a one-to-one relationship between the verb and its object. The last remaining key term is the object of the last verb after reduction. When there is an extra element before the first verb in the sentence, such as a prepositional phrase, the MAYA Semantic Technique works by removing this extra element.

B. The MAYA Semantic Technique Applied to Declarative Sentences

According to [2] a simple declarative sentence consists of NP, the subject, followed by a verb phrase (VP), the predicate. A simple VP may consist of some adverbial modifiers followed by the main verb and its complement. [2] There may be several complements in a sentence. By counting the frequency of each category, the MAYA determines the number of complements that exist in a sentence. The complement is used to determine the main verb in the sentence. By reducing the key categories until the count is one, the main complement and its corresponding verb are determined. The MAYA Semantic Technique works because the complement is always attached to its verb.

C. Conclusions from the MAYA Semantic Technique

The MAYA Semantic Technique determines the subject, verb, and object of the sentence. The subject and object of a C++ text can be a key category ("noun," "program" or "computer") or a combination of categories. The "program" category is used to represent programming concepts whereas the "computer" category is used to represent C++ keywords. C++ sentences are simple declarative sentences. Declarative sentences, whether simple or complex, state a fact, make a statement or an argument. The uses of categories in C++ text are clear. For example, key categories such as nouns, and the C++ categories such as "computer" and "program" do not become verbs or adjectives in C++ text. The MAYA Semantic Technique then gives a reduced form of the sentence (i.e., partial semantics). The output of the MAYA Semantic Technique is a representation of that sentence.

Source	Number of Sentences	Number correct	Percent of partial semantics that was correct
http://www.cee.hw.ac.uk/~pjbk/pathways/cpp1/node99.html	20	17	85
http://www.cee.hw.ac.uk/~pjbk/pathways/cpp1/node60.html	4	4	100
http://www.cee.hw.ac.uk/~pjbk/pathways/cpp1/node87.html	2	2	100
http://www.cee.hw.ac.uk/~pjbk/pathways/cpp1/node100.html	2	2	100
http://www.cee.hw.ac.uk/~pjbk/pathways/cpp1/node142.html	4	3	75

XII. Testing and Results

To develop the lexicon for C++ programming, domain text was selected from pages in two textbooks. The pages were randomly chosen. By applying the MAYA Semantic Technique to all the sentences, the necessary content of the C++ programming knowledge base was determined. The number of correct sentences is relative to the total number of sentences that the system used as input. Following the construction of the domain knowledge base the MAYA Semantic Technique was run on two pages with the following results:

Result 1

Fig. 18 The MAYA Semantic Technique has been manual tested with a page from [7]

Result 2

Source	Number of Sentences	Number correct	Percent of partial semantics that was correct
C++ Primer Plus	4	4	100

Fig. 19 The MAYA Semantic Technique has been tested with a page from [24]

The MAYA Semantic Technique was then applied to the following pages found on the World Wide Web with the results shown in Figure 20.

http://www.cee.hw.ac.uk/~pjbk/pathways/cpp1/node99.html
http://www.cee.hw.ac.uk/~pjbk/pathways/cpp1/node60.html
http://www.cee.hw.ac.uk/~pjbk/pathways/cpp1/node87.html
http://www.cee.hw.ac.uk/~pjbk/pathways/cpp1/node100.html
http://www.cee.hw.ac.uk/~pjbk/pathways/cpp1/node142.html

Fig. 20 The MAYA Semantic Technique has been tested with several web pages from [14]

Source	Number of Sentences	Number correct	Percent of partial semantics that was correct
C++ How to Program	21	17	80

These pages were found through a cursory search of the World Wide Web for C++ language content. After applying the Maya Semantic Technique, the results demonstrate a 75% to 100% accuracy rate, a high percentage. Correctness is based on the system retrieval of the subject-verb-object of the sentences.

XIII. Conclusion

The MAYA Semantic Technique employs mathematical principles to determine the partial semantics of declarative sentences. Using extended ASCII characters to represent knowledge, the MAYA technique applies mathematical /statistical methods to perform pattern matching techniques and to determine the content of a web page. Hence, the results are partial semantics and concise summarizing of the web page.

MAYA Semantic Technique currently has only been implemented with C++ programming language on texts. The content is technical, and the keywords and programming concepts are finite. Additional categories that define and describe the content of the technical and scientific texts are needed for effective implementation of this semantic technique.

The system produces partial semantics by eliminating the complementary information does not play a role in determining relevancy in the system. The partial semantic consist of the subject-verb-object of the sentence, which is needed to determine whether the information will be extracted from the web page. Currently the system compares the sentences subject and predicate from the web page with the knowledge base content to determine relevancy. Consistency is achieved by reducing redundancy of data by having the system check whether the new information from the web page is different from the data in the system's knowledge base.

This research contributes two new techniques to the field of natural language processing. The first is the use of extended ASCII characters for knowledge representation. The second

is the use of MAYA Semantic Technique to derive partial semantics for technical texts. The future applications for MAYA Semantic Technique are to extract and retrieve information from other disciplines.

ACKNOWLEDGMENT

I wish to express my gratitude to my research advisor M. Peter Jurkat, Ph.D., for his guidance, encouragement and perseverance.

Peter J. B. King of Heriot–Watt University, Edinburgh, Scotland.

REFERENCES

[1] Akman, V. (1999) *Situation Semantics as Natural Language Semantics*. (http://www.cs.bilkent.edu.tr/~akman/jour-papers/sigart/node3.html)
[2] Allen, J., (1995) *Natural Language Understanding.* (2nd ed.). Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc.
[3] Branden, C., & Tooze, J. (1991) *Introduction to Protein Structure*. New York: Garland Publishing, Inc.
[4] Brill, E. (2002) *Eric Brill Part-of-speech Tagger*. (http://www.cs.jhu.edu/~brill/)
[5] Cowie, C (2004) Semantics. The University of Sheffield, Sheffield, UK (http://www.shef.ac.uk/english/modules/ell303/docs/3)
[6] De Beaugrande, R. (1991) *Linguistic Theory: The Discourse of Fundamental Works*. New York: Longman.
[7] Deitel, H. M., & Deitel, P. J. (1998) *C++ How to Program. Upper Saddle River*, NJ: Prentice Hall, Inc.
[8] Dinneen, F. P. (1967) *An Introduction to General Linguistics*. New York: Holt, Rinehart and Winston, Inc.
[9] Einstein, A.,Lorentz, Weyl, H., & Minkowski, H., (1923) *The Principle of Relativity: A Collection of Original Memoirs on the Special and General Theory of Relativity*. (A. Sommerfeld , Notes), (W. Perrett & G. B. Jeffery, Trans.). New York: Dover Publications, Inc.
[10] Fromkin, V. A. (Ed.), Curtiss, S., Hayes, B. P., Hyams, N., Keating, P. A., Koopman, H., Munro, P., Sportiche, D., Stabler, E. P., Steriade, D., Stowell, T., Szabolcsi, A. (2000) *Linguistics: An Introduction to Linguistic Theory*. Malden, Massachusetts: Blackwell Publishers.
[11] Harris A., Korsakov M., Wakefield M., Baxter M., and Farinha D. (2003) South Bank University, London .(http://www.scism.sbu.ac.uk/inmandw/tutorials/ka/g1/)
[12] Jurafsky, D., Martin, J. H. (2000) *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, New Jersey: Prentice Hall.
[13] Jurkat, M. P. personal interview. (1999) Stevens Institute of Technology.
[14] King, Peter J. B. (1999), November. Heriot–Watt University. Edinburgh, Scotland. (http://www.cee.hw.ac.uk/~pjbk/)
[15] Langer, S. K. (1967) *An Introduction to Symbolic Logic*. New York: Dover Publications.
[16] Lyons, J. (1977) *Semantics*. Volume 1. Cambridge. Cambridge University Press.
[17] Lyons, J (1995) *Linguistics Semantics*. Cambridge. Cambridge University Press.
[18] Manning, C. D., & Schütze, H. (1999) *Foundation of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press.
[19] Matthews, P. (1997) *The Concise Oxford Dictionary of Linguistics*. Oxford. Oxford University Press
[20] Nirenburg, S., & Defrise, S. (1992) *Application-Oriented Computational Semantics*. In Michael Rosner and Roderick Johnson (Eds.). Computational Linguistics and Formal Semantics. (pp. 223-256). New York: Cambridge University Press.
[21] Partee, B. H. (1992) *Syntactic Categories and Semantic Type*. In Michael Rosner and Roderick Johnson (Eds.). Computational Linguistics and Formal Semantics. (pp. 97-126). New York: Cambridge University Press.
[22] Pinker, S. (1995) The Language Instinct. New York: Harper Perennial.
[23] Pinker, S. (2000) The Language Instinct. New York: Perennial.
[24] Prata, S. 1998. *C++ Primer Plus*. Corte Madera, CA: Mitchell Waite Group Press.
[25] Lecomte, J. INaLF/CNRS, (1998) WinBrill, http://www.cnrs.fr/ Pustejovsky, J., Boguraev, B.
[26] Russell, S., & Norvig, P. (1995) *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice Hall.
[27] Shapiro, S. C. (Ed.). (1990) *Encyclopedia of Artificial Intelligence* (Vols 1 & 2). New York: Wiley Interscience Publication, John Wiley and Son.

[i] Refer to my dissertation.