

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,900

Open access books available

146,000

International authors and editors

185M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Intelligent Information Access Based on Logical Semantic Binding Method

Rabiah A. Kadir¹, T.M.T. Sembok² and Halimah B. Zaman²

¹*Universiti Putra Malaysia /Faculty of Computer Science and IT*

²*Univeristi Kebangsaan Malaysia /Faculty of Science and IT
Malaysia*

1. Introduction

The idea of the computer system capable of simulating understanding with respect to reading a document and answering questions pertaining to it has attracted researchers since the early 1970s. Currently, the information access has received increased attention within the natural language processing (NLP) community as a means to develop and evaluate robust question answering methods. Most recent work has stressed the value of information access as a challenge in terms of their targeting successive skill levels of human performance and the existence of independently developed scoring algorithm and human performance measures. It is an exciting research implementation in natural language understanding, because it requires broad-coverage techniques and semantic knowledge which can be used to determine the strength of understanding the natural language in computer science.

In 2003, MITRE Corporation defined a new research paradigm for natural language processing (NLP) by implementing question answering system on reading comprehension. Reading comprehension offers a new challenge and a human-centric evaluation paradigm for human language technology. It is an exciting testbed for research in natural language understanding towards the information access research problem.

The current state-of-the-art development in computer-based language understanding makes reading comprehension system as a good project (Hirschman et al., 1999). It can be a valuable state-of-the-art tool to access natural language understanding. It has been proven by series of work on question answering for reading comprehension task, and it reported an accuracy of 36.3% (Hirschman et al., 1999) on answering the questions in the test of stories. Subsequently, the work of Charniak et al. (2000), Riloff & Thelen (2000), Ng et al. (2000) and Bashir et al. (2004) achieved 41%, 39.8%, 23.6% and 31.6%-42.8% accuracy, respectively. However, all of the above systems used a simple bag-of word matching, bag-of verb stem, hand-crafted heuristic rules, machine learning and advanced BOW and BOV approach. In contrast, this topic will discuss a logic representation and logical deduction approach for an inference. We aim to expand upon proposed logical formalisms towards semantic for question answering rather than just on surface analysis. Set of words, lexical and semantic clues, feature vector and a list of word token were utilized for knowledge representation in this approach.

This topic describes a method for natural language understanding that concerned with the problem of generating an automated answer for open-ended question answering processes that involve open-ended questions (ie. WHO, WHAT, WHEN, WHERE and WHY). The problem of generating an automated answer involves the context of sophisticated knowledge representation, reasoning, and inferential processing. Here, an existing resolution theorem prover with the modification of some components will be explained based on experiments carried out such as: knowledge representation, and automated answer generation. The answers to the questions typically refer to a string in the text of a passage and it only comes from the short story associated with the question, even though some answers require knowledge beyond the text in the passage. To provide a solution to the above problem, the research utilizes world knowledge to support the answer extraction procedure and broadening the scope of the answer, based on the theory of cognitive psychology (Lehnert, 1981, Ram & Moorman, 2005). The implementation used the backward-chaining deduction reasoning technique of an inference for knowledge based which are represented in simplified logical form. The knowledge based representation known as Pragmatic Skolemized Clauses, based on first order predicate logic (FOPL) using Extended Definite Clause Grammar (X-DCG) parsing technique to represent the semantic formalism.

This form of knowledge representation implementation will adopt a translation strategy which involves noun phrase grammar, verb phrase grammar and lexicon. However, the translation of stored document will only be done partially based on the limited grammar lexicon. The queries will be restricted to verb and noun phrase form to particular document. The restriction adopted in the query is appropriate, since the objective is to acquire inductive reasoning between the queries and document input. Logical-linguistic representation is applied and the detailed translation should be given special attention. This chapter deals with question answering system where the translation should be as close as possible to the real meaning of the natural language phrases in order to give an accurate answer to a question. The aim of the translation is to produce a good logical model representation that can be applied to information access process and retrieve an accurate answer. This means that logical-linguistic representation of semantic theory chosen is practically correct for the intended application.

The representation of questions and answers, and reasoning mechanisms for question answering is of concern in this chapter. To achieve a question answering system that is capable of generating the automatic answers for all types of question covered, implementation of logical semantic binding with its argument into existing theorem prover technique will describe in this chapter. Different types of questions require the use of different strategies to find the answer. A semantic model of question understanding and processing is needed, one that will recognize equivalent questions, regardless of the words, syntactic inter-relations or idiomatic forms. The process of reasoning in generating an automated answer began with the execution of resolution theorem proving. Then, the answer extraction proceeded with logical semantic binding approach to continue tracking the relevant semantic relation rules in knowledge base, which contained the answer key in skolem constant form that can be bounded. A complete relevant answer is defined as a set of skolemize clauses containing at least one skolem constant that is shared and bound to each other. The reasoning technique adopted by the system to classify answers, can be classed into two types: satisfying and hypothetical answers. Both classes were formally

distinguished based on its answers; either explicitly or implicitly as stated in the text. The goal of using logical semantic binding approach over logical forms has allow for more complex cases, such as in *Why* question where the information extracted is an implicit context from a text passage. The types of questions conducted using this approach are considered as causal antecedent, causal consequent, instrumental or procedural, concept completion, judgemental and feature specification.

The enhancement of logical-linguistic also depends on the discourse understanding from the external knowledge as an additional input in order to understand the text query and produce as its output some description or hypernyms of the information conveyed by the text. World knowledge is a knowledge about the world, that is, particularly referred to the experience or compilations of experience with other information that are not referring to a particular passage that is being asked and it would be true in real world. Real world knowledge refers to the type of knowledge from the end-user, the architectural or implementation knowledge from the software developer and other levels of knowledge as well. World knowledge is used to support the information extraction procedure and to broaden the scope of information access based on the theory of cognitive psychology (Ram & Moorman, 2005). However, several research which started in 2001, tried to exploit world knowledge to support the information extraction (Golden & Goldman, 2001; Ferro et al., 2003).

Information access task is retrieves a set of most relevant answer literal for a query is attempted. Therefore, binding are performed between the query given and the stored documents that are represented in Pragmatic Skolemize Clauses logical form. This chapter presents a comprehensive discussion of how logical semantic binding approach is practical to access the information semantically.

2. Syntax-semantic formalism

In addition to handling the semantic of a language which involves in ascertaining the meaning of a sentence, this section describes the nature of reading comprehension that includes the understanding of a story. Generally, the understanding of a document can be deciphered based on case-by-case sentences. This can be done by sentence understanding through the study of context-independent meaning within individual sentence which must include event, object, properties of object, and the thematic role relationship between the event and the object in the sentences. Based on this theory of sentence understanding, an experiment was executed based on logical linguistics and DCG was chosen as the basis of semantic translation.

2.1 Document understanding

Document understanding focused on inferential processing, common sense reasoning, and world knowledge which are required for in-depth understanding of documents. These efforts are concerned with specific aspects of knowledge representation, an inference technique, and question types (Hirschman et al. 1999; Lehnert et al. 1983; Grohe & Segoyfin 2000).

The challenge to computer systems on reading a document and demonstrating understanding through question answering was first addressed by Charniak (1972) in Dalmás et al. (2004). This work showed the diversity of both logical and common sense

reasoning which needed to be linked together with what was said explicitly in the story or article and then to answer the questions about it. More recent works have attempted to systematically determine the feasibility of reading comprehension as a research challenge in terms of targeting successive skill levels of human performance for open domain question answering (Hirschman et al. 1999; Riloff & Thelen 2000; Charniak et al. 2000; Ng et al., 2000; Wang et al. 2000; Bashir et al. 2004; Clark et al. 2005). The work initiated by Hirschman (1999), also expressed the same data set. Earlier works from years 1999 until 2000 introduced the 'bag-of-word' to represent the sentence structure. Ferro et al. (2003) innovated knowledge diagram and conceptual graph to their sentence structure respectively. This thesis, however, shall focus on the logical relationship approach in handling syntactic and semantic variants to sentence structure. This approach will be discussed thoroughly in the following sections and chapters.

The input of document understanding is divided into individual sentences. Intersentential interactions, such as reference is an important aspect of language understanding and the task of sentence understanding. The types of knowledge that are used in analyzing an individual sentence (such as syntactic knowledge) are quite different from the kind of knowledge that comes into play in intersentential analysis (such as knowledge of discourse structure).

2.1.1 Sentence understanding

A sentence can be characterised as a linear sequence of words in a language. The output desired from a sentence understander must include the event, object, properties of object, and the thematic role relationship between the event and the object in the sentence (Ram & Moorman 2005). In addition, it is also desirable to include the syntactic parse structure of the sentence. A fundamental problem in mapping the input to the output in terms of showing sentence understanding is the high degree of ambiguity in natural language. Several types of knowledge such as syntactic and semantic knowledge can be used to resolve ambiguities and identify unique mappings from the input to the desired output. Some of the different forms of knowledge relevant for natural language understanding (Allen 1995; Doyle 1997; Mahesh 1995; Mueller 2003; Dowty et al. 1981; Capel et al. 2002; Miles 1997) are as follows:

- i. Morphological knowledge – this concerns how words are constructed from more basic meaning units called morphemes. A morpheme is the primitive unit of meaning in a language. For example, the meaning of word *friendly* is derivable from the meaning of the noun *friend* and the suffix *-ly*, which transforms a noun into an adjective.
- ii. Syntactic knowledge – this concerns how words can be put together to form correct sentences and determines what structural role each word plays in the sentence and what phrases are subparts of other phrases.
- iii. Semantic knowledge – this concerns what words mean and how these meanings combine in sentences to form sentence meanings. This involves the study of context-independent meaning.
- iv. Pragmatic knowledge – this concerns how sentences are used in different situations and how its use affects the interpretation of a sentence.
- v. Discourse knowledge – this concerns how the immediately preceding sentences affect the interpretation of the next sentence. This information is especially important for interpreting pronouns and temporal aspects of the information conveyed.

- vi. World knowledge – this includes the general knowledge pertaining to the structure of the world that the language user must have in order to, for example, maintain a conversation. It includes what each language user must know about the other user's beliefs and goals.

Recognizing Textual Entities

There is various textual entities in a document that must be recognized. Following are an examples of textual entities:

Words: *was, pushed*

Phrases: *freight elevator, buffer springs*

Times: *yesterday, last week*

Places: *downtown Brooklyn, East End*

Names: *John J. Hug, Mary-Ann*

Numbers: *\$1,200, 12 inches*

These entities may be detected using various techniques. Regular expressions and pattern matching are often used (Mueller 1999; Zamora 2004; Li & Mitchell 2003). For example, in the system ThoughtTreasure developed by Mueller (1998), provides text agents for recognizing lexical entries, names, places, times, telephone numbers, media objects, products, prices, and email headers.

Anaphora

A document understanding system must resolve various anaphoric entities on the objects to which they refer (Mitkov 1994). Examples of anaphoric entities are pronouns (*she, they*), possessive determiners (*my, his*), and arbitrary constructions involving the following:

Adjectives (*the pink milk*)

Genitives (*Jim's milk*)

Indefinite and definite articles (*an elevator salesman, the shaft, the buffer springs*)

Names (*John J. Hug*)

Relative clauses (*the \$1,200 they had forced him to give them, the milk that fell on the floor*)

Anaphora resolution is a difficult problem to tackle. However, in this research, the anaphora resolution will be attained by adding world knowledge as an input to the original passage.

Commonsense Knowledge Bases

A commonsense knowledge base is a useful resource for a document understanding system. Most importantly, the commonsense knowledge base can evolve along with the document understanding system. Whenever a piece of commonsense knowledge comes in handy in the document understanding system, it can be added to the database. The database can then be expanded, thus becomes useful for the document understanding application.

The above databases have various advantages and disadvantages such as WordNet (Fellbaum 1998), which was designed as a lexical rather than a conceptual database. This

means that it lacks links between words in different syntactic categories. For example, there is no link between the noun *creation* and the verb *create*.

2.2 First-order predicate logic syntax-semantic formalism

A crucial component of understanding involves computing a representation of the meaning of sentences and texts. The notion of representation has to be defined earlier, because most words have multiple meanings known as senses (Fillmore & Baker 2000; Sturgill & Segre 1994; Vanderveen & Ramamoorthy 1997). For example, the word *cook* can be sensed as a verb and a sense as a noun; and *still* can be sensed as a noun, verb, adjective, and adverb. This ambiguity would inhibit the system from making appropriate inferences needed to model understanding.

To represent meaning, a more precised language is required. The tools to do this can be derived from mathematics and logic. This involves the use of formally specified representation languages. Formal languages are comprised of very simple building blocks. The most fundamental is the notion of an atomic symbol, which is distinguishable from any other atomic symbol that is simply based on how it is written.

2.2.1 Syntax

It is common, when using formal language in computer science or mathematical logic, to abstain from details of concrete syntax in term of strings of symbols and instead work solely with parse trees. The syntactic expressions of FOPLs consist of terms, atomic formulas, and well-formed formulas (wffs) (Shapiro 2000; Dyer 1996). Terms consist of individual constants, variables and functional terms. Functional terms, atomic formula, and wffs are nonatomic symbol structures. The atomic symbols of FOPLs are individual constants, variable, function symbols, and predicate symbols. Individual Constants comprised the following:

- i. Any letter of the alphabet (preferable early)
- ii. Any (such) letter with a numeric subscript
- iii. Any character string not containing blanks or other punctuation marks. For example, *Christopher, Columbia*.

Variables comprised the following:

- i. Any letter of the alphabet (preferably late)
- ii. Any (such) letter with a numeric subscript. For example, *x, xy, g7*.

Function Symbols comprised the following:

- i. Any letter of the alphabet (preferably early middle)
- ii. Any (such) letter with a numeric subscript
- iii. Any character string not containing blanks. For example, *read_sentence, gensym*.

Predicate Symbols comprised the following:

- i. Any letter of the alphabet (preferably late middle)
- ii. Any (such) letter with a numeric subscript
- iii. Any character string not containing blanks. For example, *noun, prep*.

Each function symbol and predicate symbol must have a particular arity. The arity need not be shown explicitly if it is understood. In any specific predicate logic language individual constant, variables, function symbols, and predicate symbols must be disjointed.

Syntax of Terms: Every individual constant and every variable are considered a term.

If f^n is a function symbol of arity n , and t_1, \dots, t_n are terms, then $f^n(t_1, \dots, t_n)$ is a (functional) term.

example:

$$\begin{aligned} & \text{free_vars}(C, \text{FreeVars}), \\ & \text{free_vars}([C_0 | Cs], \text{Fvs}, \text{FVs}) \end{aligned}$$

Syntax of Atomic Formulas:

If P^n is a predicate symbol of arity n , and t_1, \dots, t_n are terms, then $P^n(t_1, \dots, t_n)$ is an atomic formula.

example:

$$\begin{aligned} & \text{proper_noun}(\text{male}, \text{christopher}). \\ & \text{noun}(\text{bear}, \text{bears}). \end{aligned}$$

Syntax of Well-Formed Formulas (Wffs): Every atomic formula is a wffs. If P is a wff, then so is $\neg P$. if P and Q are wffs, then so are $(P \wedge Q)$, $(P \vee Q)$, $(P \Rightarrow Q)$, and $(P \Leftrightarrow Q)$. If P is a wffs and x is a variable, then $\forall x(P)$ and $\exists x(P)$ are wffs. \forall is called the universal quantifier. \exists is called the existential quantifier. P is called the scope of quantification.

Parentheses are not accounted with when there is no ambiguity, in which case \forall and \exists will have the highest priority, then \wedge and \vee will have higher priority than \Rightarrow , which, in turn will have higher priority than \Leftrightarrow . For example, $\forall xP(x) \wedge \exists yQ(y) \Leftrightarrow \neg P(a) \Rightarrow Q(b)$ will be written instead of $((\forall x(P(x)) \wedge \exists y(Q(y))) \Leftrightarrow (\neg P(a) \Rightarrow Q(b)))$.

Every concurrence of x in P , not on the scope of some occurrence of $\forall x$ or $\exists x$, is said to be free in P and bound in $\forall xP$ and $\exists xP$. Every occurrence of every variable other than x that is free in P is also free in $\forall xP$ and $\exists xP$. A wff with at least one free variable is called open, no free variables are called closed, and an expression with no variables is called ground.

Syntactic Category: Below, is a syntactic category of English fragment covered by DCG that is given by the set SynCat (Partee 2006; Partee 2001):

$$\text{SynCat} = \{S, \text{NP}, \text{VP}, \text{DET}, \text{CNP}, \text{ProperN}, \text{ADJ}, \text{REL}, \text{CN}, \text{TV}, \text{IV}, \text{PP}\}$$

The elements of SynCat are symbols representing the English categories as follows:

S: sentences

NP: noun phrases

VP: verb phrases

DET: determiners

CNP: common noun phrases

ProperN: proper nouns

ADJ: adjectives

REL: relative clauses

CN: common nouns

TV: transitive verbs

IV: intransitive verbs

PrepP: prepositional phrase

2.2.2 Semantic

Although the intensional semantics of a FOPL depend on the domain being formalized, and the extensional semantics depend also on a particular situation, specification on the types of entities is usually given as the intensional and the extensional semantic of FOPL expressions.

The usual semantic of FOPL assumes a Domain, D , of individuals, function on individuals, sets of individuals, and relations on individuals. Let I be the set of all individuals in the Domain D .

Semantic of Atomic Symbols

Individual Constants:

If a is an individual constants, $[a]$ is some particular individual in I .

Function Symbols:

If f^n is a function symbol of arity n , $[f^n]$ is some particular function in D ,

$$[f^n]: I \times \dots \times I \rightarrow I \text{ (} n \text{ times)}$$

Predicate Symbols:

If P^1 is a unary predicate symbol, $[P^1]$ is more particular subset of I . If P^n is a predicate symbols of arity n , $[P^n]$ is some particular subset of the relation $I \times \dots \times I$ (n times).

Semantic of Ground Terms

Individual Constants:

If a is an individual constant, $[a]$ is some particular individual in I .

Functional Terms:

If f^n is a function symbol of arity n , and t_1, \dots, t_n are ground terms, then $[f^n(t_1, \dots, t_n)] = [f^n]([t_1], \dots, [t_n])$.

Semantic of Ground Atomic Formulas

- i. If P^1 is unary predicate symbol, and t is a ground term, then $[P^1(t)]$ is True if $[t] \in [P^1]$, and False otherwise.

- ii. If P^n is an n -ary predicate symbol, and t_1, \dots, t_n are ground terms, then $[P^n(t_1, \dots, t_n)]$ is True if $\langle [t_1], \dots, [t_n] \rangle \in [P^n]$, and False otherwise.

Semantic of Wffs

- i. If P is a ground wff, then $[\neg P]$ is True if $[P]$ is False, otherwise, it is False.
- ii. If P and Q are ground wffs, then $[P \wedge Q]$ is True if $[P]$ is True and $[Q]$ is True, otherwise, it is False.
- iii. If P and Q are ground wffs, then $[P \vee Q]$ is False if $[P]$ is False and $[Q]$ is False, otherwise, it is True.
- iv. If P and Q are ground wffs, then $[P \Rightarrow Q]$ is False if $[P]$ is True and $[Q]$ is False, otherwise, it is True.
- v. If P and Q are ground wffs, then $[P \Leftrightarrow Q]$ is True if $[P]$ and $[Q]$ are both True or both False, otherwise, it is False.
- vi. $[\forall xP]$ is True if $[P\{t/x\}]$ is True for every ground term, t . Otherwise, it is False.
- vii. $[\exists xP]$ is True if there is some ground term, t such that $[P\{t/x\}]$ is True. Otherwise, it is False.

Semantic Types of FOPL

Every English expression of a particular syntactic category is translated into semantic expression of a corresponding type. The semantic types are defined as in Table 1.

Syntactic Category	Semantic Type	Expressions
S	t	sentences
ProperN	e	names (<i>Chris</i>)
CN(P)	$e \rightarrow t$	common noun phrases (<i>cat</i>)
NP	e	"e-type" or "referential" NPs (<i>Chris, the president</i>)
	$e \rightarrow t$	NPs as predicates (<i>an animal, a president</i>)
ADJ(P)	$e \rightarrow t$	predicative adjectives (<i>pretty, big</i>)
REL	$e \rightarrow t$	relative clauses (<i>who(m) read the book</i>)
VP, IV	$e \rightarrow t$	verb phrases, intransitive verbs (<i>read the book, is big</i>)
TV	$\langle e, e \rangle \rightarrow t$	transitive verbs (<i>read, lives</i>)
is	none	temporary treatment: pretend it is not there
DET	$e \rightarrow t$ to e	universal quantifier (<i>the</i>)
	$e \rightarrow t$ to $e \rightarrow t$	existential quantifier (<i>a</i>)

Table 1. Semantic Types of Syntactic Category in FOPL

where:

e is a type, representing object of sort entity

t is a type, representing truth values

Based on Table 1, some of the compositional semantics are as follows:

S: Denotes a truth value, relative to an assignment of the values to free variables

NP: Is of two kinds. First, referential NP formed with definite article *the* is of type e and denotes an individual, as in *The boy writes*. Second, Predicate NP formed with indefinite

article a is of type $e \rightarrow t$ and denotes a set, as in *Christopher is a boy*. There are lexical NPs of the referential kind, including proper nouns (*George, Robin*) and indexed pronoun (*he_i*) which will be interpreted as individual variable x_i .

CN, CNP, ADJ, VP, IV, REL: All of type $e \rightarrow t$, one-place predicates, denoting sets of individuals. For this type, the parser will freely go back and forth between sets and their characteristic function, treating them as equivalent.

TV: Is a type of $\langle e, e \rangle \rightarrow t$, a function from ordered pairs to truth values, i.e. the characteristic function of a set of ordered pairs. A 2-place relation is represented as a set of ordered pairs, and any set can be represented by its characteristic function.

DET (a): Form predicate nominals as an identity function on sets. It applies to any set as argument and gives the same set as value. For example, the set of individuals in the model who are student, $\langle a \text{ student} \rangle = \{ \{ a \} \mid (\{ \mid \text{student} \mid \}) = \{ \mid \text{student} \mid \}$

DET (the): Form e-type NPs as the *iota* operator, which applies to a set and yields an entity if its presuppositions are satisfied, otherwise it is undefined. It is defined as follows:

$\{ \mid \iota \alpha \mid \} = d$ if there is one and only one entity d in the set denoted by $\{ \mid \alpha \mid \}$

$\{ \mid \iota \alpha \mid \}$ is undefined otherwise

For example: the set of animals who Chris love contains only Pooh, then *the animal who Chris loves* will denote Pooh. If Chris loves no animal or loves more than one animal, then *the animal who Chris loves* is undefined, i.e. has no semantic value.

Semantic Representation of English Expression in FOPL

Table 2 shows the semantic representation or syntax-semantic formalism that represents a number of simple basic English expressions and phrases, along with a way of representing the formula in Prolog.

Syntactic Category	Semantic Representation	As written in Prolog
<i>Christopher</i> (PN)	logical constant <i>Christopher</i>	christopher
<i>animal</i> (CN)	1-place predicate $(\lambda x)animal(x)$	$X^{animal}(x)$
<i>young</i> (ADJ)	1-place predicate $(\lambda x)young(x)$	$X^{young}(x)$
<i>young animal</i> (CN with ADJ)	1-place predicate joined by 'and' $(\lambda x)young(x) \wedge animal(x)$	$X^{young}(X), animal(X)$
<i>writes</i> (TV)	2-place predicate $(\lambda y)(\lambda x)writes(x,y)$	$Y^{X^{writes}(X,Y)}$
<i>read</i> (IV)	1-place predicate $(\lambda x)read(x)$	$X^{read}(X)$
<i>is an animal</i> (Copular VP)	1-place predicate $(\lambda x)animal(x)$	$X^{animal}(x)$
<i>with</i> (PrepP)	1-place predicate $(\lambda y)(\lambda x)with(x,y)$	$Y^{X^{with}(X,Y)}$

Table 2. Representation of Simple Words and Phrases

The basic expression *animal* and *young*, is a category of CN and ADJ, are translated into predicate $(\lambda x)\text{animal}(x)$ and $(\lambda x)\text{young}(x)$ respectively. However, the word *young* is considered as a property, not as a thing. This has to do with the distinction between sense and reference. A common noun such as *owl* can refer to many different individuals, so its translation is the property that these individuals share. The reference of *animal* in any particular utterance is the value of x that makes $\text{animal}(x)$ true.

These are different with phrases, such as verbs which require different numbers of arguments. For example, the intransitive verb *read* is translated into one-place predicate $(\lambda x)\text{read}(x)$. Meanwhile, a transitive verb such as *writes* translates to a two-place predicate such as $(\lambda y)(\lambda x)\text{writes}(x,y)$. The copula (*is*) has no semantic representation. The representation for *is an animal* is the same as for *animal*, $(\lambda x)\text{animal}(x)$.

Basic expressions can be combined to form complex expressions through unification process, which can be accomplished by arguments on DCG rules. The following shows the illustration of combining several predicates in the N^1 by joining them with \wedge (and) symbol (Covington 1994). From

$$\begin{aligned}\text{young} &= (\lambda x)\text{young}(x) \\ \text{smart} &= (\lambda x)\text{smart}(x) \\ \text{animal} &= (\lambda x)\text{animal}(x)\end{aligned}$$

then, the complex expression will be presented as:

$$\text{young smart animal} = (\lambda x)(\text{young}(x) \wedge \text{smart}(x) \wedge \text{animal}(x))$$

DCG rules for the lexicon entries for the particular words:

$$\begin{aligned}\text{adj}(X \wedge \text{young}(X)) &\rightarrow [\text{young}]. \\ \text{adj}(X \wedge \text{smart}(X)) &\rightarrow [\text{smart}]. \\ \text{adj}(X \wedge \text{green}(X)) &\rightarrow [\text{green}]. \\ \text{noun}(X \wedge \text{animal}) &\rightarrow [\text{animal}]. \\ \text{noun}(X \wedge \text{cat}) &\rightarrow [\text{cat}].\end{aligned}$$

The syntactic and translation rules of DCG are equivalent to the rules defined in PS. For the PS rules, the semantic of the whole N^1 is as follows:

$$n1(\text{Sem}) \rightarrow n(\text{Sem}).$$

and below is the rule that combines an adjective with an N^1 :

$$n1(X \wedge (P,Q)) \rightarrow \text{adj}(X \wedge P), n1(X \wedge Q).$$

Through these implementation rules, basic English expressions are combined to form complex expressions, and at the same time translated into FOPL expressions using Prolog unification process. The implementation rule for the determiner in natural language corresponds to the quantifiers in formal logic. The determiner (DET) can be combined with a common noun (CN) to form a noun phrase. The determiner or quantifier \exists normally goes with the connective \wedge , and \forall with \rightarrow . The sentence *An animal called Pooh* contains quantifier and its semantic representation is presented as $(\exists x)(\text{animal}(x) \wedge \text{called}(x, \text{Pooh}))$. In this case, Prolog notation is written as $\text{exist}(X, \text{animal}(X), \text{called}(X, \text{Pooh}))$.

3. Knowledge representation

Knowledge representation is the symbolic representation aspects of some closed universe of discourse. They are four properties in a good system for knowledge representations in our domain, which are representation adequacy, inferential adequacy, inferential efficiency, acquisition efficiency (Mohan, 2004). The objective of knowledge representations is to make knowledge explicit. Knowledge can be shared less ambiguously in its explicit form and this became especially important when machines started to be applied to facilitate knowledge management. Knowledge representation is a multidisciplinary subject that applies theories and techniques from three other fields (Sowa, 2000); logic, ontology, and computation.

Logic and Ontology provide the formalization mechanisms required to make expressive models easily sharable and computer aware. Thus, the full potential of knowledge accumulations can be exploited. However, computers play only the role of powerful processors of more or less rich information sources. It is important to remark that the possibilities of the application of actual knowledge representation techniques are enormous. Knowledge is always more than the sum of its parts and knowledge representation provides the tools needed to manage accumulations of knowledge.

To solve the complex problems encountered in artificial intelligent, it needs both a large amount of knowledge and some mechanisms for manipulating that knowledge to create solution to new problems. Putting human knowledge in a form with which computers can reason it is needed to translate from such 'natural' language form, to some artificial language called symbolic logic. Logic representation has been accepted as a good candidate for representing the meaning of natural language sentences (Bratko, 2001) and also allows more subtle semantic issues to be dealt with. A complete logical representation of open-ended queries and the whole text of passages need an English grammar and lexicon (Specht, 1995; Li, 2003). The output requested for reading comprehension task from each input English phrase must include the event, object, properties of object, and the thematic role relationship between the event and the object in the sentence (Ram & Moorman, 2005).

The translation strategy involves noun phrase grammar, verb phrase grammar and lexicon which are built entirely for the experiment purposes. However, the translation of stored passages will only be done partially based on the limited grammar lexicon. The queries will be restricted to verb and noun phrase form. The restriction adopted in the query is appropriate, since the objective of the reading comprehension task in this research, is to acquire deductive reasoning between the queries and passage input. Therefore, this can be done using verb and noun phrase. Some evidence has been gathered to support this view (Ferro et al., 2003; Bashir et al., 2004).

This work deals with question answering system where the translation should be as close as possible to the real meaning of the natural language phrases in order to give an accurate answer to a question. The query given and the stored passages are represented in PragSC logical form. In general, the translation of the basic expressions or English words into semantic templates are based on their syntactic categories as shown in Table 3, where, X stand for object CN, Y stand for object CN or ADJ, 'predicate' stands for the English word, λ stand for exists or all, and 'app-op' stand for & or \rightarrow .

Categories	Template Forms
CN	[X predicate(X)]
TV	[[X A], Y A & predicate(X, Y)]
IV	[[X A] A & predicate(X)]
ADJ	[X predicate(X)]
DET	[[X A], [X C] λ (X, A app-op C)]
Prep	[[X A], Y A & predicate(X, Y)]
AUX	Temporary treatment as in FOPL: pretend it is not there

Table 3. Syntax-semantic formalism of english fragment

The syntax and semantic formalism to define the notion of representation due to shows the meaning of a sentence. A new logical form, known as PragSC has been proposed for designing an effective logical model representation that can be applied to question answering process and retrieve an accurate answer. The main advantage of logical representation in this problem is its ability to gives names to the constituents such as noun phrase and verb phrase. This means that it recognizes a sentence as more than just a string of words. Unlike template and keyword approach, it can describe recursive structure, means the longer sentence have shorter sentences within them. Figure 1 illustrates the example of English phrase (an animal called pooh) translation:

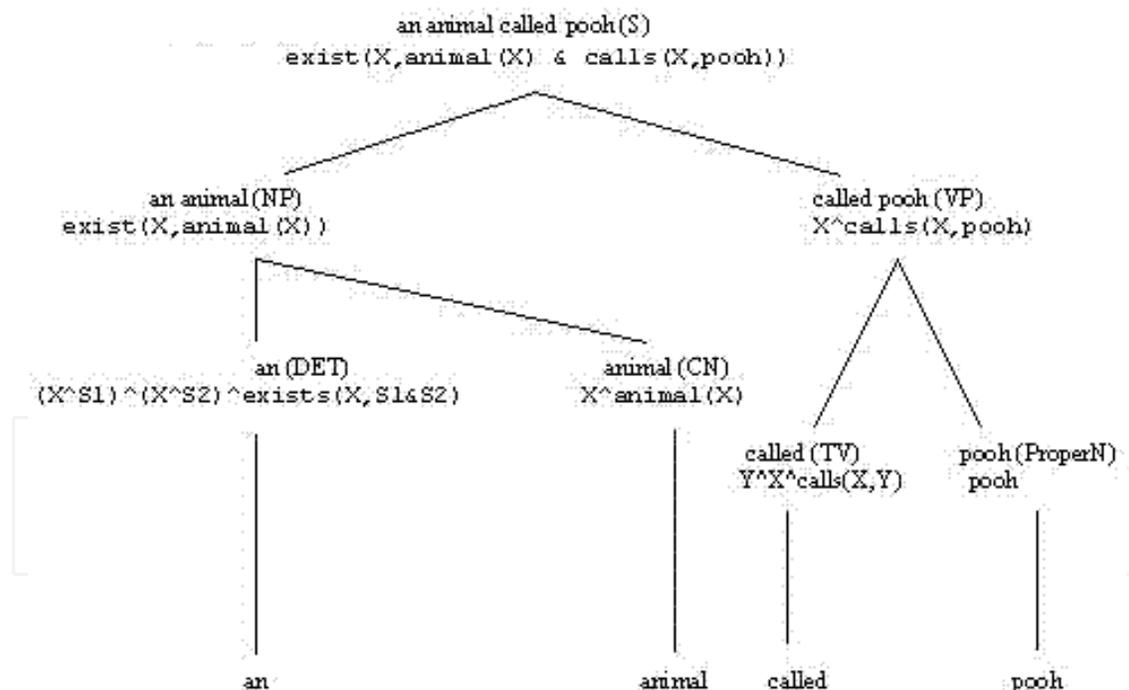


Fig. 1. Semantic tree

Each Natural language text is directly translated into PragSC form which can be used as a complete content indicator of a passage or query. The passages and queries are processed to form their respective indexes through the translation and normalization process which are composed of simplification processes. The similarity values between the passage and query indexes are computed using the skolemize clauses binding of

resolution theorem prover technique. This representation is used to define implication rules for any particular question answering and for defining synonym and hypernym words.

A query is translated into its logical representation as documents are translated. This representation is then simplified and partially reduced. The resulting representation of the query is then ready to be proven with the passage representation and their literal answers are retrieved. The proving is performed through uncertain implication process where predicates are matched and propagated, which finally gives a literal answer value between the query and the passage. In the following section, a more detailed description of the query process and its literal answer value will be discussed.

4. Logical semantic binding inference engine

Work on open-ended question answering requires sophisticated linguistic analysis, including discourse understanding and deals with questions about nearly everything, and not only relying on general ontologies and world knowledge. To achieve a question answering system that is capable of generating the automatic answers for all types of question covered, implementation of skolemize clauses binding with its argument into existing theorem prover technique is introduced. Automated theorem proving served as an early model for question answering in the field of AI (Wang et al., 2000). Whereas, skolemize clauses binding approach over logical forms has allow for more complex cases, such as in Why question where the information extracted is an implicit context from a text passage. Skolemize clauses binding approach relates how one clause can be bound to others. Using this approach, the proven theorem need only to determine which skolem constant can be applied to, and valid clauses will be produced automatically.

Skolemize clauses binding is designed to work with simplified logical formula that is transformed into Pragmatic Skolem Clauses form. The basic idea is that if the key of skolemize clause match with any skolemize clauses in knowledge base, then both clauses are unified to accumulate the relevant clauses by connecting its normalize skolem constant or atom on the subject side or the object side of another. The normalize skolem constant or atom is a key for answer depending on the phrase structure of the query. Given a key of skolemize clause in negation form and a set of clauses related in knowledge base in an appropriate way, it will generate a set of relevant clauses that is a consequence of this approach. Lets consider the example of English query Why did Chris write two books of his own? to illustrate the idea of skolemize clauses binding.

Example: Why did Chris write two books of his own?

Key skolemize clause:

$\sim \text{write}(\text{chris}, \text{g15})$.Unification:

$\sim \text{write}(\text{chris}, \text{g15}) \text{ :- } \text{write}(\text{chris}, \text{g15})$

Key of answer (Object):

g15Set of relevant clauses:

```
two(g15)
book(g15)
his(g9)
own(g9)
of(g15,g9)
write(chris,g15)
two(g15)
book(g15)
famous(g18)
be(likes(tells(g15,it)),g18)
```

The example considered that `write(chris,g15)` is the key skolemize clauses. `g15` is the key of answer that is used to accumulate the relevant clauses through linking up process either to its subject side or object side.

Implementation of skolem clauses binding is actually more complicated when the clauses contain variables. So two skolemize clauses cannot be unified. In this experiment, the operation involves “normalization” of the variables just enough so that two skolemize clauses are unified. Normalization is an imposition process of giving standards atom to each common noun that exists in each input text passage which was represented as variable during the translation process. The skolem clauses normalization involves X-DCG parsing technique that has been extended with functionality of bi-clausifier. The detail of X-DCG parsing technique has been explained in chapter 5. Skolem constants were generated through the first parsing process. Then, the process of normalization was implemented in second parsing, which is a transformation process identifying two types of skolem constant to differentiate between quantified (f_n) and ground term (g_n) variable names.

Whereas, binding is a term within this experiment, which refers to the process of accumulating relevant clauses by skolem constant or atom connected to any clauses existing in knowledge base. Each skolemize clause is conceived as connected if each pair of clause in it is interrelated by the key answer which consists of a skolem constant or atom. The idea that it should be specific is based on coherent theory which deals with this particular set of phenomena, originated in the 1970s, based on the work in transformational grammar (Peters & Ritchie, 1973; Boy, 1992).

This work was conducted to solve the problem by connecting the key of an answer that has been produced through resolution theorem prover. Skolemize clauses binding technique gives the interrelation of skolemize clauses that could be considered as a relevant answer by connecting its key of answer. To establish this logical inference technique, Figure 2 illustrates the inference engine framework.

An answer is literally generated by negating a query and implementing skolemize clauses binding. This will enable a resolution theorem prover to go beyond a simple “yes” answer by providing a connected skolem constant used to complete a proof. Concurrently, a semantic relation rule is also specified in pragmatic skolemize clauses as a knowledge base representation. In the example provided, this can be seen as binding process proceeds. If the semantic relation rule being searched contains rules that are unified to a question through its skolem constant, the answers will be produced. Consider the following sample as a

semantic relation rule used as an illustration that was originally based on a children passage entitled "School Children to Say Pledge" from Remedia Publications.

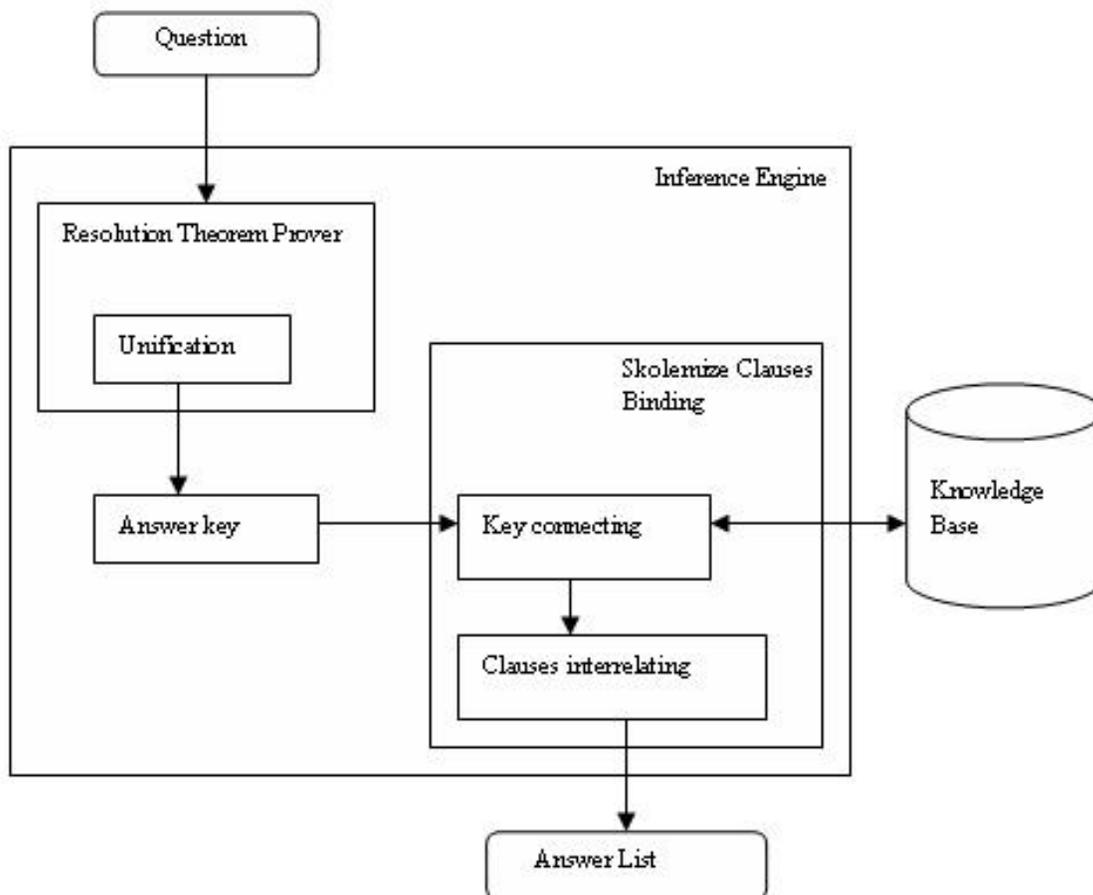


Fig. 2. The architecture of an inference engine framework

Semantic relation rules in PragSC form:

```

cl([pledge(f25)],[])
cl([young(g37)],[])
cl([people(g37)],[])
cl([proud(g38)],[])
cl([feels(g37,g38)],[])
cl([makes(f25,g37)],[])
cl([writes(r(frances & bellamy),f25)],[])

```

Given above is the simple semantic relation rules, and the question Why did Frances Bellamy write the pledge?, then, the following logical form of question are produced.

$$\sim \text{pledge}(f25) \# \sim \text{writes}(r(\text{frances \& bellamy}),f25) \# \text{answer}(f25)$$

Based on the above representation, $f25$ and $r(\text{frances \& bellamy})$ is unified with the semantic relation rules in knowledge base;

$$\sim \text{pledge}(f25) :- \text{pledge}(f25)$$

$\sim \text{writes}(\text{r}(\text{frances \& bellamy}), \text{f25}) :- \text{writes}(\text{r}(\text{frances \& bellamy}), \text{f25})$

then, bind both entities to any relevant semantic relation rule to find the answer.

- a. $\text{cl}([\text{pledge}(\text{f25}), []])$
- b. $\text{cl}([\text{young}(\text{g37}), []])$
- c. $\text{cl}([\text{people}(\text{g37}), []])$
- d. $\text{cl}([\text{proud}(\text{g38}), []])$
- e. $\text{cl}([\text{feels}(\text{g37}, \text{g38}), []])$
- f. $\text{cl}([\text{makes}(\text{f25}, \text{g37}), []])$
- g. $\text{cl}([\text{writes}(\text{r}(\text{frances \& bellamy}), \text{f25}), []])$

The skolemized clauses (a) to (g) are a collection of answer sets that are unified to the question given because each clause is bound with at least one skolem constant. The semantic relation rule base indicates that $\text{r}(\text{frances \& bellamy})$ is bound to clause (g), meanwhile f25 (pledge) is bound to clause (a) and (g). The system continue tracking any relevant semantic relation rules in knowledge base, which contain skolem constant f25 that can be bounded. In this case clause (f) is picked out. Clause (f) gives more binding process by another skolem constant, g37 , represent young people predicate. The process of skolem constant binding was retained until there are no skolem clauses which can be bounded. It is a process of accumulating of relevant clauses by *skolem constant* (x) or *atom* connected to any clauses existing in knowledge base.

$$x \rightarrow P(x, x_1) \wedge P(x_1, x_2) \wedge \dots \wedge P(x_{n-1}, x_n) \wedge P(x_n) \quad (1)$$

The example is motivated by showing what happened when the facts, $\text{r}(\text{frances \& bellamy})$ and pledge are bound to other clauses or semantic relation rules. Then, the resulting answer is:

$\text{makes}(\text{f25}, \text{g37})$
 $\text{young}(\text{g37})$
 $\text{people}(\text{g37})$
 $\text{feels}(\text{g37}, \text{g38})$
 $\text{proud}(\text{g38})$

All the skolemized clauses were considered as a set of answer that is relevant to the question, and they may be the best information available. Another examples are shown in Table 2. Each example begins with part of a collection of semantic rules in knowledge base, represented in skolemized clauses. In this research, a question Q is represented as a proposition, and a traditional proof initiated by adding the negation of the clause form of Q to a consistent knowledge base K . If an inconsistency is unified, then skolemized clauses binding process proceed to find the relevant answer.

4.1 Relevant answer

A relevant answer to a particular question can be generally defined as an answer that implies all clauses to that question. Relevance for answers has been defined as unifying the skolem constant by the question. In a rule base consisting solely of skolem constants, the

unifying of a single skolem constant to a question would be considered a relevant answer. When rules are added, the experiment becomes more complicated. When taxonomic relationship is represented in a rule base, a relevant answer can be defined as an interconnection of all clauses that unify and bind the same skolem constants. Table 4 depicts two examples illustrate the skolemized clauses binding process to extract relevant answer.

	Example 1	Example 2
Semantic relation rules (K)	cl([now(g1)],[]) cl([new(f1)],[]) cl([faster(f1)],[]) cl([way(f1)],[]), cl([sents(g1,f1)],[]) cl([now(g1)],[]) cl([end(r(pony & express),g1)],[])	cl([two(g46)],[]) cl([book(g46)],[]) cl([own(his)],[]) cl([writes(chris,g46)],[]) cl([famous(g52)],[]) cl([be(like (tells(g46,it)),g52)],[])
Proposition (Q)	~ end(r(pony & express),g1) # answer(g1)	~ two(g46) # ~ book(g46) # ~ writes(chris,g46) # answer(g46)
Unifying process	~ end(r(pony & express),g1) :- end(r(pony & express),g1)	~ two(g46) :- two(g46) ~ book(g46) :- book(g46) ~ writes(chris,g46) :- writes(chris,g46)
SCB Key connecting	g1 connecting: now(g1) mail(g1) sents(g1,f1)	g46 connecting: two(g46) book(g46) be(like (tells(g46,it)),g52)
SCB Clauses interrelating	now(g1) mail(g1) new(f1) faster(f1) way(f1) sents(g1,f1)	two(g46) book(g46) famous(g52) be(like(tells(g46,it)),g52)

Table 4. Example of question answering process

The first example in Table 4, g1 is considered as a skolem constant to be unified to a skolemized clause in knowledge base, ~ end(r(pony & express),g1) :- end(r(pony & express),g1). Then g1 binds to any skolemized clauses consisting of the same skolem constant, and tracks all possible skolemized clauses in knowledge base by binding skolem constant exists, f1, until all skolem constants bindings are complete. The relevant answer consists of several clauses that are bound by g1. The output is as follows:

```
sents(g1,f1).
now(g1).
mail(g1).
new(f1).
faster(f1).
way(f1).
```

Same as the first example, this second example recognised $g46$ as a skolem constant to be unified to a skolemized clauses in knowledge base which involve more than one clauses to be unified, $\sim \text{two}(g46) :- \text{two}(g46); \sim \text{book}(g46) :- \text{book}(g46); \sim \text{writes}(\text{chris},g46) :- \text{writes}(\text{chris},g46)$. Then, $g46$ binds to any skolemized clauses consisting of the same skolem constant, and tracks all possible skolemized clauses in knowledge base by binding skolem constant exists, $g52$, until all skolem constants bindings are complete. The relevant clauses are as follows:

$\text{two}(g46)$
 $\text{book}(g46)$
 $\text{famous}(g52)$
 $\text{be}(\text{like}(\text{tells}(g46,\text{it})),g52)$.

Throughout this experiment, providing information in a form of pragmatic skolemized clauses is just a method to collect the keywords for relevant answers. The issues related to the problem of providing an answer in correct English phrases can be considered another important area of research in question answering. In this research this problem has been considered, but thus far it has taken the form of observations rather than formal theories. This represents an area for further research interest.

5. Intelligent information access

This topic aims is to extract some relevant answers which are classified into satisfying and hypothetical answers. When the idea of an answer is expanded to include all relevant information, question answering may be viewed as a process of searching for and returning of information to a questioner that takes different places in time. As one of the most challenging and important processes of question answering systems is to retrieve the best relevant text excerpts with regard to the question, Ofoghi et al. (2006) proposed a novel approach to exploit not only the syntax of the natural language of the questions and texts, but also the semantics relayed beneath them via a semantic question rewriting and passage retrieval task. Therefore, in our experiment, we used logics description to provide a natural representation and reasoning mechanism to answer a question which is a combination of resolution theorem prover and a new approach called skolemize clauses binding.

On the other hand, external knowledge sources are added in order to give more understanding of text and produce some descriptions of the information conveyed by the text passages. External knowledge sources consist of two components with different roles of usage and motivation. First, world knowledge is used to solve the outstanding problem related to the ambiguity introduced by anaphora and polysemy. Meanwhile, in the second component, hypernyms matching procedure constitute the system in looking for the meaning of superordinates words in the question given. The purpose of this component is to produce a variety of answers based on different ways on how it is asked. This thesis has clearly demonstrated their importance and applicability to question answering, including their relationship to the input passage in natural language. In particular, this thesis is focused on providing detailed formal definition of world knowledge.

Situating a query as a concept in a taxonomic hierarchy makes explicit the relationship among type of questions, and this is an important part of intelligent intelligent extraction. A

logical technique solves a constraint satisfaction problem by the combination of two different methods. Logical reasoning applied an inference engine to extract an automatic answer. Logical technique exploits the good properties of different methods by applying them to problems they can efficiently solve. For example, search is efficient when the problem has many solutions, while an inference is efficient in proving unsatisfiability of overconstrained problems.

This logical technique is based on running search over a set of variables and inference over the other ones. In particular, backtracking or some other form of search is executed with a number of variables; whenever a consistent partial assignment over these variable is found, an inference is executed on the remaining variables to check whether this partial assignment can be extended to form a solution based on logical approach. This affects the choice of the variables evaluated by the search. Indeed, once a variable is evaluated, it can be effectively extracted from the knowledge base, restricting all constraints it is involved with in its value. Alternatively, an evaluated variable can be replaced by a skolem constant, one for each constraint, all having a single-value domain. This mixed technique is efficient if the search variables are chosen in a manner where duplicating or deleting them turns the problem into one that can be efficiently solved by inference.

6. Discussion and conclusion

To appreciate fully the significance of the findings of this research, it helps to firstly understand the level of scientific rigor used to guide the formation of conclusions from the research. The experiments are considered complete when the expecting results or findings replicate across previous research and settings. Findings with a high degree of replicability are finally considered as incontrovertible findings and these form the basis for additional research. Each research study within this research domain network usually follows the most rigorous scientific procedures.

The study does not embrace any a priori theory, but represent the linguistic knowledge base into logical formalisms to build up the meaning representation and enforce syntactic and semantic agreements that include all information that are relevant to a question. In a true scientific paradigm, the study is tested in different behaviour or condition which involve two kinds of external knowledge sources. This contrasts with the usual nature of previous researches in the same domain, where none was ever tested against all four conditions as in this study. The detail of the research works and experiences are as follows:

- **Logical Interpreter Process.** The interpreter process, whether it be for translation or interpreting, can be described as decoding the meaning of the source text and re-encoding this meaning in the target representation. In this experiment the target representation is in simplified logical model. To decode the meaning of a text, the translator must first identify its component "interpreter units," that is to say, the segments of the text to be treated as a cognitive unit. A interpreter unit may be a word, a phrase or even one or more sentences. Behind this seemingly simple procedure lies a complex cognitive operation. To decode the complete meaning of the source text, the interpreter must consciously and methodically interpret and analyze all its features. This process requires thorough knowledge of the interpreter, grammar, semantics, syntax, dictionary, lexicons and the like, of the source language. The interpreter needs

the same in-depth knowledge to re-encode the meaning in the target language. In fact, in general, interpreters' knowledge of the target language is more important, and needs to be deeper than their knowledge of the source language.

The interpreter is a domain-independent embodiment of logical inference approach to generate a clauses form representation. The translation process is guided by a set of phrase structure rules of the sentence and build a tree structure of sentence. The rules mean: An S can consist of an NP followed by a VP. An NP can consist of a D followed by an N. A VP can consist of a V followed by an NP, and etc. This set of rules is called a Definite-Clause Grammar (DCG) as shown below:

S :- NP, VP
NP :- D, N
VP :- V, NP

The parsing process is like left-right top-down parsers, DCG-rule parsers go into a loop when they encounter a rule of the form. Each position in the tree has labels, which may indicate procedure to be run when the traversal enters or leaves that position. The leaves of the tree will be words, which are picked out after morphological processing, or pieces of the original text passage. In the latter case, the interpreter looks up the phrase structure in the lexicon dictionary to find realization for the words that satisfied the lexical items. Below is shown an example of lexical items.

D(a, singular).
N(animal, animals).
V(amaze, amazes, amazed, amazed, amazing, amazes).

The result is a new logical form representation of phrase structure tree, possibly with part(s) of the original text passage. In this way, the entire text passage is gradually translated into logical form as shown below.

alive(_36926 ^ isa(r(christopher & robin),_36926))
& well(_36926 ^ isa(r(christopher & robin),_36926))
exists(_46238,((pretty(_46238) & home(_46238))
& calls(_46238,r(cotchfield & farm))) & lives(chris,_46238))

After got a way of putting logical formula into a nice tidy form, an obvious thing to investigate was need a way of writing something in clausal form known as Pragmatic Skolemize Clauses (PragSC). PragSC form is a collection of clauses with at most one unnegated literal. The logical formula must turns out into PragSC form, to work with logical inference approach as proposed. The interpreter does some additional work in translation process, therefore, some modification to its was required. Before PragSC can be generated, it is required to generate a new unique constant symbol known as Skolem Constant using multi-parsing approach. The first parsing used to generate skolem constant, introducing two types of skolem constant to differentiate between quantified (f_n) and ground term (g_n) variable names. Meanwhile, the second parsing was implemented an algorithm to convert a simplified logical formula into PragSC form.

```

cl([alive(g34)],[]).
cl([isa(r(christopher & robin),g34)],[]).
cl([well(g35)],[]).
cl([isa(r(christopher & robin),g35)],[]).

cl([pretty(f29)],[]).
cl([home(f29)],[]).
cl([calls(f29,r(cotchfield & farm))],[]).
cl([lives(chris,f29)],[]).

```

- **Identifying Inference Engine Methodology.** In this experiment, the inference procedure has to identify the type to generate a relevant answer. The inference procedure is a key component of the knowledge engineering process. After all preliminary information gathering and modeling are completed queries are passed to the inference procedure to get answers. In this step, the inference procedure operates on the axioms and problem-specific facts to derive at the targeted information. During this process, inference is used to seek out assumptions which, when combined with a theory, can achieve some desired goal for the system without contradicting known facts. By seeking out more and more assumptions, worlds are generated with non-contradicting knowledge.

In inference process, implementation of skolemize clauses binding with its argument into existing theorem prover technique is introduced. The answer literal enables a resolution refutation theorem prover to keep track of variable binding as a proof proceeds. Resolution refutation can be thought as the bottom-up construction of a search tree, where the leaves are the clause produced by knowledge base and the negation of the goal. For example, if the question asked has the logical form $\exists y P(x, y)$, then a refutation proof is initiated by adding the clause $\{\neg P(x, y)\}$ to the knowledge base. When the answer literal is employed, the clause $\{\neg P(x, y), ANSWER(x)\}$ is added instead. The x in the answer literal ($ANSWER(x)$) will reflect any substitutions made to the x in $\neg P(x, y)$, but the $ANSWER$ predicate will not participate in (thus, will not effect) resolution. Then, the inference process preceded using skolemize clauses binding approach relates how one clause can be bound to others. For example, if the key of skolemize clause (x) match with any skolemize clauses in knowledge base, then both clauses are unified to accumulate the relevant clauses by connecting its normalize skolem constant or atom on the subject side or the object side of another, formulated as $x \rightarrow P(x,x1) \wedge P(x1,x2) \wedge \dots \wedge P(xn-1,xn) \wedge P(xn)$.

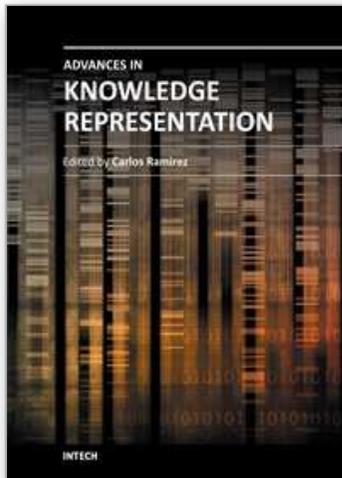
7. References

- Allen, J. 1995. *Natural Language Understanding*. 2nd Ed. Redwood City, CA: Benjamin/Cummings.
- Bashir, A., Kantor, A., Ovesdotter C. A., Ripoche, G., Le, Q., & Atwell, S. (2004). Story Comprehension. Retrieved April 11, 2005 from <http://12r.cs.uiuc.edu/~danr/Teaching/CS598-04/Projects/First/Term2.pdf>
- Capel, A., Heaslip, L., & Williamson, D. 2002. *English Grammar*. H. P. 1990. 1st Ed. Westerhill Road, Bishopbriggs, Glasgow G64 2QT: HarperCollins Publishers.
- Charniak, E. 1972. *Toward a Model of Children's Story Comprehension*. PhD Thesis. Institute of Technology, Massachusetts.

- Charniak, E., Altun, Y., Braz, R.d.S., Garret, B., Kosmala, M., Moscovich, T., Pang, L., Pyo, C., Sun, Y., Wy, W., Yang, Z., Zeller, S., and Zorn, L. (2000). Reading Comprehension Programs in an Statistical-Language-Processing Class. In Proceeding of the ANLP/NAACL 2000 Workshop on Reading Comprehension Test as Evaluation for Computer-Based Language Understanding Systems, (pp. 1-5).
- Clark, C., Hodges, D., Stephan, J., & Moldovan, D. 2005. Moving QA Towards Reading Comprehension Using Context and Default Reasoning. American Association for Artificial Intelligence, CA.
- Dalmas, T., Leidner, J.L., Webber, B., Grover, C., & Bos, J. 2004. Generating Annotated Corpora for Reading Comprehension and Question Answering Evaluation. <http://remote.science.uva.nl/~mdr/NLPQA/07dalmas-et-al.pdf> [2 April 2004].
- Dowty, D. R., Wall, R. E., & Peters, S. 1981. *Introduction to Montague Semantics*. Dordrecht, Holland: D. Reidel Publishing Co.
- Doyle, P. 1997. Natural Language Understanding: AI Qual Notes. <http://www1.cs.columbia.edu/nlp/paper.html> [14 November 2003].
- Dyer, C. R. 1996. Lecture Notes: First-Order Logic (Chapter 8 - 9). <http://www.cs.wisc.edu/~dyer/cs540/notes/fopc.html> [10 March 2004].
- Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. Cambridge: MIT Press.
- Ferro, L., Greiff, W., Hirschman, L., & Wellner, B. 2003. Reading, Learning, Teaching. <http://www.mitre.org/news/events/> [2 April 2004].
- Fillmore, C. J., & Baker, C.F. 2000. Frame Semantic for Text Understanding. *International Journal of Lexicography* 16 (3): 363-366.
- Golden, T. M., & Goldman, S.R. 2001. Development and Evaluation of an Automatized Comprehension Assessment Tool. 0113369. University of Texas at Dallas.
- Grohe, M., & Segoyfin, L. 2000. On First-Order Topologies Queries. *15th Annual IEEE Symposium on Logic in Computer Science (LICS '00)*, pp. 349 - 361
- Hirschman, L., Light, M., Breck, E., & Burger, J.D. (1999). Deep Read: A Reading Comprehension System. Proceeding of the 37th Annual Meeting of the Association for Computational Linguistics, (pp. 325 - 332).
- Lehnert, W. G. (1981). A Computational Theory of Human Question Answering. In A. K. Joshi, Weber, B.J., & Sag, I.A. *Elements of Discourse Understanding*, (pp.145-176). Cambridge: Cambridge University Press.
- Lehnert, W., Micheal, D., Johnson, P., Yang, C.J., & Harley, S. 1983. BORIS - an Experiment in In-Depth Understanding of Narrative. *Journal of Artificial Intelligence* 20 (1): 55-68.
- Mahesh, K. 1995. *Syntax-Semantic Interaction in Sentence Understanding*. PhD Thesis. Georgia Institute of Technology, Georgia.
- Miles, W. S. 1997. Natural Language Understanding. WSM Information Systems Inc.
- Mitkov, R. 1994. An Integrated Model for Anaphora Resolution. *Proceedings of the 15th Conference on Computational Linguistics*, pp. 1170 -1176
- Mueller, E. T. 2003. Story Understanding through Multi-representation Model Construction. *Proceeding of HLT-NAACL 2003 Workshop.*, pp. 46 - 53.
- Ng, H. T., Teo, L.H., & Kwan, J.L.P. (2000). A Machine Learning Approach to Answering Questions for Reading Comprehension Tests. Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000), (pp. 124-132).

- Ram, A., & Moorman, K. (2005). Towards a theory of reading and understanding. In *Understanding Language Understanding: Computational Models of Reading*, Cambridge: MIT Press.
- Riloff, E., & Thelen, M. (2000). A Rule-based Question Answering System for Reading Comprehension Tests. Proceeding of ANLP/NAACL 2000 Workshop on Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems, (pp. 13 - 19).
- Shapiro, S. C. 2000. Propositional, First-Order and Higher-Order Logics: Basic Definitions, Rules of Inference, and Examples. In *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language* April: 379 - 395
- Sturgill, D. B., & Segre, A. M. 1994. Using Hundreds of Workstations to solve First-order Logic problems. *12th National Conf. on Artificial Intelligence*, pp. 187-192
- Vanderveen, K. B., & Ramamoorthy, C. V. 1997. Anytime Reasoning in First-Order Logic. *Proc. of the 9th Int. Conf. on Tools with Artificial Intelligence (ICTAI '97)*, pp. 142 - 148

IntechOpen



Advances in Knowledge Representation

Edited by Dr. Carlos Ramirez

ISBN 978-953-51-0597-8

Hard cover, 272 pages

Publisher InTech

Published online 09, May, 2012

Published in print edition May, 2012

Advances in Knowledge Representation offers a compilation of state of the art research works on topics such as concept theory, positive relational algebra and k-relations, structured, visual and ontological models of knowledge representation, as well as detailed descriptions of applications to various domains, such as semantic representation and extraction, intelligent information retrieval, program proof checking, complex planning, and data preparation for knowledge modelling, and a extensive bibliography. It is a valuable contribution to the advancement of the field. The expected readers are advanced students and researchers on the knowledge representation field and related areas; it may also help to computer oriented practitioners of diverse fields looking for ideas on how to develop a knowledge-based application.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Rabiah A. Kadir, T.M.T. Sembok and Halimah B. Zaman (2012). Intelligent Information Access Based on Logical Semantic Binding Method, *Advances in Knowledge Representation*, Dr. Carlos Ramirez (Ed.), ISBN: 978-953-51-0597-8, InTech, Available from: <http://www.intechopen.com/books/advances-in-knowledge-representation/intelligent-information-access-based-on-logical-semantic-binding-method>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen