Improvement of Document Understanding Ability through the Notion of Answer Literal Expansion in Logical-linguistic Approach

Rabiah A.K., Sembok, T.M.T, Halimah, B.Z.

Abstract: - Document undestanding offer interesting alternative to the kinds of special-purpose, carefully constructed evaluations that have driven many recent research in language understanding. It involves the process of reading a specific text document and answer the questions about it, to demonstrate one's understanding of the document by returning exact phrase answers. This research aims to implement proposed logical formalisms by expanding the notion of answer literal for understanding task such as question answering. This paper modify the skolem arguments to broaden the notion of answer literal to all context of question that conducted, including universal quantifier and ground term. There are two symbols, f_n represents the quantified variable names, while g_n represents ground term variable names. The expanding of the notion of answer literal enables the document to be tested by all context of question, including universal quantified and ground term variables. Both answers link to the concept of capability that is considered in this experiment.

Key-Words: - Document Understanding, Linguistic Query, Logical Approach

1 Introduction

Research concerning a topic related to natural language understanding usually will only make clear its relationship to syntax, semantics, and pragmatics of the document domain for natural linguistic of computer science. The move of information technology towards language understanding to show how words relate to each other within a sentence, how the sentences relate to each other, and the organization or narrative flow of the document. Open-ended question is a tool to evaluate the ability of document understanding.

This paper discusses related works on natural language understanding, followed by semantically and pragmatically-accented approach in the context of formal grammar and linguistic semantic. The experimental shows the emphasizing the notion of literal answer from all contexts of questions can be counter-productive in extracting the exact answer. The broadening of the notion of literal answer enables the question answering can be applied to all context of question, including universal quantified and ground term variables.

Firstly, effective activity is to broaden the representation of input text of the document that should be considered as important in the natural language understading system. The introduction of the notion of the document achieved the objective of the study in representing all context of the input text using two variable names. This notion has been further emphasized and refined in many versions of endogenous growth theory along this experiment. The originator of the new idea of broadening the notion of text representation also emphasized the role answering capability in this light, seeing its existence as a prerequisite to the accumulation of document understanding activities.

Secondly, the activity broaded the set of literal answer should be considered as the intrinsic value. The literal answer enhances the ability of question answering in producing the relevant answer. It is of value in becoming a member of a community of the relevant answer. This second activity relates the recent theorisation of the importance of skolemize constant as a key for the relevant literal answer.

The two activities associated with different uses of the term input and output. In this different context, the application of question answering carries different meanings. The activities describe in greater details the aspects of the notion of answer literal, presenting schematic observation about the logical requirement of the application in order to consider the sense in which the application clarifies the understanding.

Finally, expanding the notion of answer literal including all context of the question conducted as strategies for answer recognition resulted in a greater number of understanding abilities that taught the system to produce more relevant answers.

2 Document Understanding

Document understanding focused on inferential processing, common sense reasoning, and world knowledge which are required for in-depth understanding of documents toward meaning extraction. Meaning extraction is an emerging technology that identifies elements of information and concepts contained within documents and document repositories, and surfaces combinations of these informative elements and concepts that imply meaning in the context of the business, professional, or technical purpose of the search process [1].These efforts are concerned with specific aspects of knowledge representation, inference technique, and question types [2].

More recent works have attempted to systematically determine the feasibility of document understanding as a research challenge in terms of targeting successive skill levels of human performance for open domain question answering [3, 4]. Earlier works initiated by Hirschman in year 1999 [5] until 2000 introduced the 'bag-of-word' to represent the sentence structure. Then, Ferro et al. (2003) innovated knowledge diagram and conceptual graph to their sentence structure respectively. This paper, however, shall focus on the logical relationship approach in handling syntactic and semantic variants to sentence structure in a document.

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).

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 [6]. 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 such as morphological, syntactic, semantic, pragmatic, discourse and world knowledge [7]. Problems of natural language understanding is an area at the heart of AI, which are highlighted in question answering.

The challenge to computer systems on reading a document and demonstrating understanding through question answering was first addressed by Charniak (1972) in Dalmas et al. (2004) [8]. 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 document and then to answer the questions about it.

3 Question Answering

Question answering (QA) is an activity in getting relevant information or a process of any information exchange that involves queries and the data that is not fully structured [9]. Question answering has been relatively narrowly focused on the task of searching and returning as answers of an individual that satisfy [10]. Question answering has been approached from a number of different perspectives. Cognitive-science-based approaches to question answering are concerned with trying to simulate human question answering.

Several QA systems were developed as vehicles for natural language understanding research. Two of the most famous QA systems of that time are BASEBALL and LUNAR, both of which were developed in the 1960s. Both QA systems were very effective in their chosen domains. Further restricted-domain QA systems were developed in the following years. The common feature of all these systems is that they had a core database or knowledge system that was hand-written by experts of the chosen domain.

Another two of the most famous systems that included QA abilities are SHRDLU and ELIZA [11, 12. SHRDLU simulated the operation of a robot in a toy world (the "blocks world"), and it offered the possibility to ask the robot questions about the state of the world. Again, the strength of this system was the choice of a very specific domain and a very simple world with rules of physics that were easy to encode in a computer program. ELIZA, in contrast, simulated a conversation with a psychologist. ELIZA was able to converse on any topic by resorting to very simple rules that detected important words in the person's input. It had a very rudimentary way to answer questions, and on its own it led to a series of chatterbots such as the ones that participated in the annual Loebner prize.

In late 1990s, the annual Text Retrieval Conference (TREC) included a question-answering track which has been running until present. Systems participating in this competition were expected to answer questions on any topic by searching a corpus of text that varied from year to year. This competition fostered research and development in open-domain text-based question answering.

Research in the area of open-domain question answering generates a lot of interest, both from the NLP community and the end-users of this technology, either lay users or professional information analysts. There are several purposes on getting relevant information with a different types of knowledge domain. Generally, the ability to ask questions is central to processes of reasoning, understanding and learning toward its knowledge domain. Question arises from an interaction between the interests and goals of the questioner, and the information provided by the knowledge domain.

Question answering is a complex task that needs a formal theory and well-defined evaluation methods [13, 14]. Several theories have been developed earlier in the context of NLP or cognitive sciences. It is still in the interest of question answering research to revitalize research in semantics NLP, such that one can better understand the question, the context in which they are posed, and deliver and justify answers in the context [15]. Therefor, work on open-domain question answering requires sophisticated linguistic analysis, including discourse understanding and deals with questions about nearly everything, and can only rely on general ontologies. The representation of questions and answers, and reasoning mechanisms for question answering are the concerns of research in knowledge representation and reasoning [16, 17].

4 Knowledge Representation

Knowledge representation is the symbolic representation aspects of some closed universe of discourse. Knowledge representation and reasoning are traditional areas within artificial intelligence. In the modern society they are underlying building blocks in various kinds of information systems and networks. Knowledge representation and reasoning are also central themes in cognitive science and epistemology. Relevant questions include how we know what we know, how we can make useful inferences, and how we can use computers in models and applications of knowledge representation and reasoning. Traditional models have been based on predicate logic, semantic networks and other symbolic representations.There are four properties in a good system for knowledge representations in our domain and they are representation adequacy, inferential adequacy, inferential efficiency, and acquisition efficiency. The objective of knowledge representation is to make knowledge explicit.

Knowledge Representation can be defined as the application of logic to the task of constructing computable models of some domain [18, 19]. Logic provide the formalization mechanisms required to make expressive models easily sharable and computer aware. This means that the full potential of knowledge accumulation can be exploited. However, computers only play the role of powerful processors with different levels of richness in information sources. Logic representation has been accepted as a good entity for representing the meaning of natural language sentences [20], and allows more subtle semantic issues to be dealt with.

4.1 Translation Strategy

Translation rules are relatively simple because each of them is supposed to match the whole list of words. The output of a translation rule is a list of atoms which, when converted back into character strings and concatenated, will give the appropriate simplified form of logical-linguistic. The first of these rules handle the 'quit' command that the user will use to exit from the program. The procedure that applies the translation rule will simply find a rule that applies to the input, then execute a cut, or complain if no rule applicable.

To present a document into a simplified form of logical-linguistic, it is necessary to encode the syntactic and semantic aspect of each sentence. The parser recognizes two types of semantic entities: predicate and names, and its predicate arguments relation to give the relationship of these entities. It returns error message on receiving ill-formed input. An input is considered ill-formed if it contains one of this condition: (i) unknown words – are words that are not predefined in lexicon, and these include misspelled words; (ii) non-covered lexicon dictionary – the structure of the lexicons is not covered by the lexicon-dictionary implemented, even though it is grammatically correct; and (iii) illegal grammatically syntactic structure – the structure of the input is grammatically wrong. To describe the meaning of natural language utterances, a précised way of describing the information that they contained is needed. It relies on the logical model and set theory, both of which are precisely defined knowledge base.

Consider a simple formula such as `lives(chris, england)`, represent '*Chris lives in England*'. This formula shows a part of a logical language. A logical model consists of an Entity (E), which is the set of individual people and things that can be talked about, plus a Semantic function (S) which gives a relation onto entities. This model has two important advantages. First, it assigns meaning to all parts of every formula, rather than just assigning truth values to a complete sentence. Second, a logical model works with knowledge bases without making any claims about the real world as a whole. This is important because it corresponds closely to computer manipulation of a database.

4.2 Simplified Logical Translation

Simplified form of logic is derived from the syntactic parse of the text input and each lexicon in the text will recognize two types of semantic entities: nouns and verbs. The first thing to be noted is that names are logical constant (‘*Chris*’ = `chris`), but common nouns, and noun with adjective are predicates (‘*children*’ = `(λx)children(x)`). An adjective, such as ‘small’ is considered a property, not an entity. This has to do with the distinction between sense and reference. A name refers to only one individual, thus the translation is directed to a logical constant. But a common noun such as ‘*children*’ can refer to many different individuals, so its translation is the property that these individuals share. The reference of ‘*children*’ in any particular utterance is the value of `x` that makes `children(x)` true.

Secondly, note that different verbs require different numbers of arguments. The intransitive verb ‘*barked*’ translates to a one-place predicate `(λx)(barked(x)`. A transitive verb translates to a two-place predicate `(λy)(λx)(cuts(x,y)`.

These arguments are filled in, step by step, as we progress up from common noun to NP, from verb to VP, and then S. The following examples are used to serve an illustration:

- “*At noon, two small children cut a ribbon*.”

```
noon(x1 ^ at(x1)) & two(x2 :
(small(x2) & children(x2)) &
exists(x3,ribbon(x3) &
cuts(x2,x3))
```

- “*The ribbon was made from paper.*”

```
exists(x4,ribbon(x4) & paper(x5
^ makes(x4,x5))
```

4.3 Skolem Constant Generation

Before PragSC can be generated, it is required to generate a new unique constant symbol known as Skolem Constant [25]. Each logic expression involves predicate, functions and quantifier, so that the generation of skolem constant implements a common algorithm to convert a formula into clausal form [21]. This work proposed an alteration to its skolem function where by expanded the notion of literal answer during parsing.

This first parsing, the transformed formula and the list of variables have been introduced by universal and existential quantifier, and ground term. Skolem function makes use of two new predicates. Predicate `gensym` must be defined such that the goal `gensym(X,Y)` causes `Y` to be instantiated to a new atom built up from the atom `X` and a number. This is used to generate skolem constant that have not been used before. The second new predicate mentioned is `subst`. Here it is required for `subst(V1, V2, F1, F2)` to be true if the result of substituting `V2` for `V1` every time it appears in the formula `F1` is `F2`.

```
subst(X, Sk, exists(Y,P), exists(Y,P1)) :- !,
              subst( X, Sk, P, P1 ).
subst(X, Sk, (P & Q),(P1 & Q1)) :- !,
              subst( X, Sk, P, P1 ),
              subst( X, Sk, Q, Q1 ).
subst(X, Sk, P, P1) :- functor(P,F,N).

gensym(Root, Atom) :-
       get_num(Root, Num),
       name(Root, Name1),
       integer_name(Num, Name2),
       append(Name1, Name2, Name),
       name(Atom, Name).
get_num(Root, Num) :-
       retract(current_num(Root, Num1)), !,
       Num is Num1+1,
       asserta(current_num(Root, Num)).
get_num(Root, 1) :-
       asserta(current_num(Root, 1)).
```

In the process of transformation, the normalization of the skolem constants are applied to all variable names. We identified two types of skolem constant to differentiate between quantified (f_n) and ground term (g_n) variable names. The following shows the use of f_n and g_n which stand for skolem constant in clausal form for each variable names.

```
cls(two, g9).
cls(small, g9).
cls(children, g9).
cls([ribbon, f55).
cls([paper, g10).
```

```
cls(pretty, f3).
cls(home, f3).
cls(three, g4).
cls(old, g4).
cls(year, g4).
cls(poem, f4).
```

Each skolem constant that are generated will be stored in the list of normalization clauses skolem constant for the second parsing process.

4.4 PragSC Transformation

Based on the research problem, before the resolution theorem prover can be applied, a set of simplified formula is required to be converted into what is known as clausal form. This section explains the process of transforming the simplified logical formula into clausal form, called PragSC. This transformation is a second parsing, whereas the step is the same as the first parsing which implemented an algorithm to convert a simplified logical formula into clausal form. However, since the skolem function has been modified, instead of generating a new skolem constant symbol, it will retrieve an atom that was already built up in the first parsing.

```
skolem(Pred(X:P), Pred(F)&P2, Vars ) :- !,
            getatom( Pred, F ),
            Sk =..[F|Vars],
            subst( X, Sk, P, P1 ),
            skolem_v2( P1, P2, Vars ).

getatom(Noun, Atom) :-
            (cls(Noun, Const) ->
                  name(Const, ListTemp),
                  name(Atom, ListTemp))
                ;
                  gensym_v2(g, Atom)).
```

The result of this parsing is a set of PragSC, which as knowledge base representation that can be applied to logical reasoning in the context of natural language understanding.

5 Logical Inference Process

To achieve better document understanding that is capable of generating the automatic answers for all types of question covered, implementation of logical-linguistic approach called skolemize clauses binding (SCB) with its notion skolem constant expansion into existing theorem prover technique is introduced. 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 questions from each document are chosen to measure how well the system can understand the input document.

Skolemize clauses binding approach is considered as an inference technique that is used to provide explicit and implicit answer to the questions given by considering a theorem to be proven as a question. The goal of using SCB 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 general applicability of exact answers to natural language text document may be examined with respect to a sample of Lehnert's question classes [23]. Lehnert's work is question classification scheme that still considered the basis of question classification in this current work on document understanding purposes.

5.1 Theory of SCB

This approach is implied to question answering and aims at the creation of flexible extraction which accept natural language question in English and generate relevant answer literal that contain information extracted from a document. SCB 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 (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 $x \rightarrow P(x,x1) \wedge P(x1,x2) \wedge \ldots \wedge P(xn\text{-}1,xn) \wedge P(xn)$. 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 *When was the ribbon cut?* and *Why did Chris write two books of his own?* to illustrate the idea of skolemize clauses binding.

Example 1: *When was the ribbon cut?*

Key skolemize clause:

```
~cuts(f2,g1).
```

Unification:

```
~cuts(f2,g1):- cuts(f2,g1)
```

Key of answer (Object): `g1`

Set of relevant clauses:

```
ribbon(f2).
noon(g1).
cuts(f2,g1).
```

Example 2: *Why did Chris write two books of his own?*

Key skolemize clause:

```
~write(chris,g15).
```

Unification:

```
~write(chris,g15):-
   write(chris,g15)
```

Key of answer (Object): `g15`

Set 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)
```

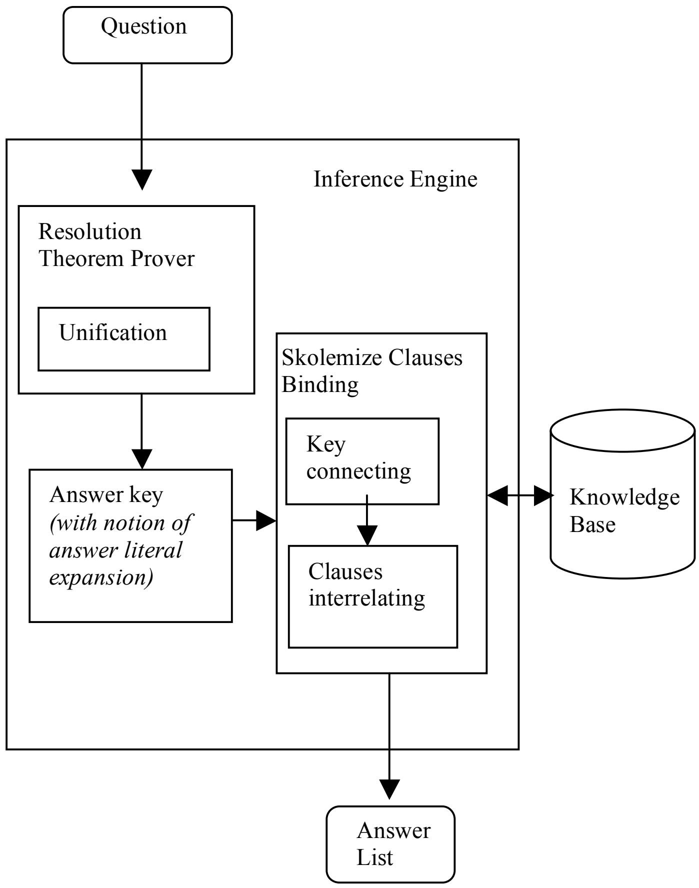

Fig. 1 The Architecture of Inference Engine Framework

Both examples considered that `cuts(f2,g1)` and `write(chris,g15)` are the key skolemize clauses. *g1* and *g15* are the keys of answer that are used to accumulate the relevant clauses through linking up process either to its subject side or object side. Skolemize clauses binding technique gave 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, Fig. 1 illustrates the inference engine framework.

The question is represented in logical form and using resolution theorem prover to produce an answer key. Skolemize Clauses binding consists of two components. First is the *key connecting* component, which is a key matching procedure. It takes a skolem constant or atom, called an answer key as an input. The answer key is an expanded notion of answer literal including universal quantified and ground term variables to broaden the extraction process. Then, the procedure returns a semantic relation rules represented in skolemize clauses form. The second component is *clauses interrelating* where all relevant skolemize clauses in knowledge base will be collected starting with semantic relation rule that was introduced by the first component. Conclusively, the list of skolemize clauses are produced in the form of a list of answers. All the skolemize clauses were considered as a set of answer that is relevant to the question, and they may be the best information available.

6 Experimental Setup

The theories described in this paper have been developed and tested using logical reasoning techniques. A logical reasoning technique includes some changes and addition to the component such as:

- the quantifier for the answer set – modify the skolem arguments including universal quantifier and ground term. There are two symbols, f_n represents the quantified variable names, while g_n represents ground term variable names. Both variables names used to all context of question that conducted in this experiment to improve the document understanding ability.
- the logical inference engine – implemention of new inference of question answering called skolemize clauses binding (SCB) into existing resolution theorem prover technique. SCB module is considered as an inference technique that is used to provide explicit and implicit answer to the questions given by considering a theorem to be proven as a question. SCB implementation enable a resolution theorem prover to go beyond a simple "yes" answer by providing a connected skolem constant used to complete a proof. If the semantic relation rule being searched contains rules that are unified to a question through its skolem constant, the answers will be produced.
- the phrase structure used to represent a clause – develop a simplified form of logical knowledge representation that is designed based on First Order Logic (FOL). The simplified form of logical-oriented model is known as Pragmatic Skolemised Clauses Representation (PragSC). It includes the event, object, properties of object, and the thematic role relationship between the event and the object in the sentence.

Fig. 2 shows the model of document understanding process to illustrate the whole system implementation. In order to perform this experiment, a familiar document from Remedia Publications data set has been used. Each document has five *wh* questions (*who, what, when, where and why*) with several classification such as causal antecedent, causal consequent, instrumental or procedural, concept completion, judgemental, and feature specification.

Logical reasoning techniques with the broaden notion of answer literal is a complete inference engine for knowledge base containing PragSC representation. Providing information in a form of pragmatic skolemized clauses together with the expanded notion of literal answer is just a method to collect the relevant answers. Proof start with the required goal, then resolution theorem prover is applied to provide the answer key by keeping track of variable as a proof proceeds. Then, the answer extraction proceed with skolemize clauses binding approach to continue tracking any relevant semantic relation rules in knowledge base, which contain the answer key in skolem constant form that can be bounded.

In addition, this experiment used WordNet to provide a group of English words into sets of hypernyms called *hypsets*, contain list of hypenyms words, general definitions, and records the various semantic relations in which one word is the hypernym of another [26]. It looking at the output of a parser and taking all the terms linked by constructions such as *X and other Y*; *X* could be considered a possible hyponym of *Y*. The process is by taking hypernyms pairs from WordNet and finding many noun-noun pairs from a parsed corpus. The procedure train a classifier to select those pairs of words that have a high probability of being hypernym pairs given the constructions which the terms in the corpus.

The advantage of this experiment is allowing the system to apply additional knowledge such as hypernyms matching procedure, supplied by wordnet for question answering in document understanding activity that can broaden the scope of answers rather than just expanding the notion of answer literal.

6.1 Implementation Control

Since this experiment conducted a logical reasoning technique, there are two considerations related to the control of the inference engine. The first is determining when the resolution process should halt. Resolution theorem prover is designed to halt when an empty clause is generated, which makes sense when the goal of resolution is to find a proof. Therefore, when a literal answer is employed,

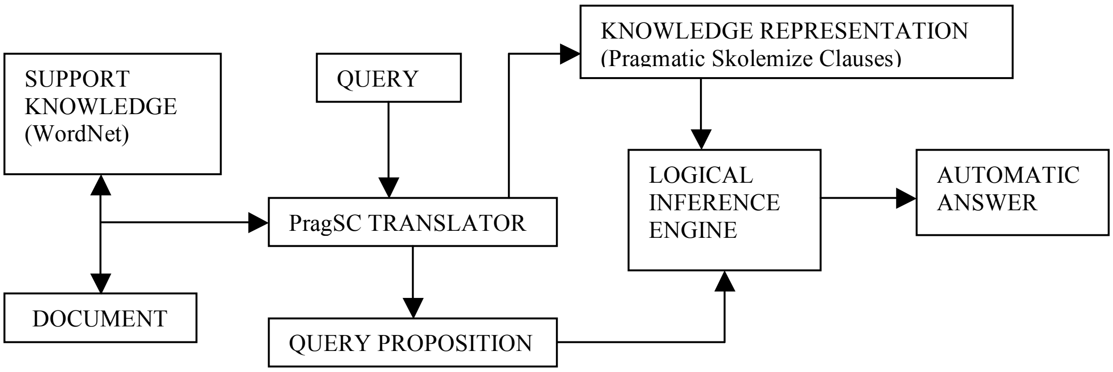

Fig. 2 Model of Document Understanding Process

a proof is associated with a clause containing onlyliteral answers. Since this experiment's interest is in finding all relevant answers (whenever possible), stopping when the empty clause is generated is not appropriate. Instead, reasoning ishalted when the set of support is empty. When the number of automatic answers is infinite, some alternative mechanism must be available to halt the reasoning process because the set of support will never be empty. The problem of an infinite reasoning process will appear when there is no proof. Prolog provides the semicolon option after the completion of a proof, and this gives the user option of looking for an additional proof.

The second control issue known as skolemize clauses binding is the search strategy on which clauses should be chosen at any point to be resolved. Clauses are stored in knowledge base. In employing the search strategy, at least one clause to be resolved must be selected from the question clauses through its skolem constant binding.

The original version of the theorem prover incorporate unit preference, sorting the clauses in question clauses and knowledge base sets from smallest to largest length. The length of a clause is defined as the number of literals in the clause. Rather than relying on the built-in sorting of clauses, the user may elect to manually select the clauses to resolve at each step of resolution instead. By giving an additional proof to the original resolution theorem prover, the reasoning process allows to conduct search strategies and preferences may be examined.

7 Result Analysis

This section presents the results obtained by the approach chosen and a comparative analysis of the performance of answer extraction. In order to show a desired answer for a variety of *wh* questions, the feasibility of document understanding evaluation is illustrated. This experiment observed the effect of the logical reasoning technique, on the performance of automatic answer extraction with emphasizing the notion of literal answer from all contexts of questions.

This experiments adopted previous automatic results [5, 22, 24], as a comparative study for precision analysis in automatic answer extraction. However, the best results among these previous research produced by Bashir et al. [24] were chosen as the benchmark throughout the analysis. Performance levels of automatic answer extraction were measured based on 5*Wh* questions with two conditions of experiment. The results obtained for each type of question are shown separately in Table 1.

Table 1 Result Analysis of Wh Question Answering

	Who	**What**	***When***	**Where**	**Why**
Benchmark	0.465	0.432	0.558	0.476	0.191
Without Expanded Notion of Answer Literal	0.530	0.417	0.496	0.478	0.226
Expanded Notion of Answer Literal	0.682	0.650	0.670	0.717	0.439

As results indicated in Table 1, the performance of automatic answer extraction varied substantially across question types. In average, the system performed the best on *WHERE* questions, achieving 71.7% correctly and it causes 24.1% increment over the benchmark. Question *WHY*, which had the most complicated scheme for handling the automatic answer extraction, performed the worst answer precision and it reached only 43.9%. In particular, *WHY* questions proved to be the most difficult of the question types throughout the whole experiment, either in this current work or previous works which produced the lowest percentage of correct answers extraction throughout the experiments. However, in this experiment it can be considered to have produced a better result with an average of 24.8% growth.

8 Conclusion

We presented a strategy of answer detection resulted in document understanding abilities that taught the system to produce more relevant answers. This work clearly shows that emphasizing the notion of literal answer from all contexts of questions can be counter-productive in extracting the exact answer. The expanding notion of literal answer includes universal quantified and ground term variables. Both answers link to the concept of capability that is considered in this experiment.

The first effective activity is to broaden the representation of input text of document that should be considered as important in the natural language understanding. The introduction of the notion of the document achieved the objective of the work in representing all context of the input text. This notion has been further emphasized and refined in many versions of endogenous growth theory along this study. The originator of the new idea of broadening the notion of text representation also emphasized the role answering capability in this light, seeing its existence as a prerequisite to the accumulation of document understanding system.

The second activity may be to broaden the set of literal answer that should be considered as the intrinsic value. The literal answer is not of value only because it enhances the ability of question answering in producing the relevant answer. It is of value in becoming a member of a community of the relevant answer. This second activity relates the recent theorisation of the importance of skolemize constant as a key for the relevant literal answer.

The two activities as described are associated with different uses of the term input and output. The activities describe in greater details the aspects of the notion of answer literal, presenting schematic observation about the logical requirement of the application in order to consider the sense in which the application clarifies the understanding.

For future work, we would like to explore ways of modeling negative answer literal, more finely controlled term extraction, such as restricting a text corpora around each query term, to get more precise answer terms. We are also interested in investigating how the union of anaphora resolvers and disambiguators would affect results.

References:

[1] David, S. How does Meaning Extarction Actually Work? http://blogs.northernlight.com/ceo/2009/01/06/how-does-meaning-extraction-actually-work/ 2009. Available online (April 2009)

[2] David, S. How does Meaning Extraction Actually Work? http://blogs.notherlight.com/ceo/2009/01/06/2009 Available online (April 2009).

[3] Bashir, A., Kantor, A., Ovesdotter C. A., Ripoche, G., Le, Q., & Atwell, S. 2004. Story Comprehension. http://12r.cs.uiuc.edu/~danr/Teaching/CS598-04/Projects/First/Term2.pdf Available online (11 April 2005).

[4] Clark, C., Hodges, D., Stephan, J., & Moldovan, D. Moving QA Towards Reading Comprehension Using Context and Default Reasoning. *American Association for Artificial Intelligence,* CA, 2005.

[5] Hirschman, L., Light, M., Breck, E., & Burger, J.D. Deep Read: A Reading Comprehension System. *Proc. of the 37th Annual Meeting of the Association for Computational Linguistics*, 1999, pp. 325 – 332.

[6] Ram, A., & Moorman, K. Towards a theory of reading and understanding. In *Understanding Language Understanding: Computational Models of Reading*, Cambridge: MIT Press, 2005.

[7] Mueller, E. T. Story Understanding through Multi-representation Model Construction. *Proceeding of HLT-NAACL 2003 Workshop.*, 2003, pp. 46 – 53.

[8] Dalmas, T., Leidner, J.L., Webber, B., Grover, C., & Bos, J. Generating Annotated Corpora for Reading Comprehension and Question Answering Evaluation. http://remote.science.uva.nl/~mdr/NLPQA/07dalmas-et-al.pdf Available online (2 April 2004).

[9] Galitsky, B. *Natural Language Question Answering System*. 2. Adelaide: Advanced Knowledge International Pty Ltd, 2003.

[10] Burhans, D. T. A Question Answering Interpretation of Resolution Refutation. Ph.D Dissertation. Faculty of the Graduate School, State University of New York, Buffalo, 2002.

[11] Weizenbaum, J. Eliza: A computer program for the study of natural lan-guage communication between man and machine. *Journal of Communication, ACM*, Vol.9, No.1, 1966, pp. 36-45.

[12] Martincic, C. J. 2003. QUE: An Expert System Explanation Facility That Answer "Why Not" Types of Questions. *Journal CSC,* Vol.19, No.1, 2003, pp. 336 - 348.

[13] Nyberg, E., Mitamura, T., Carbonell, J., Callan, J., Collins-Thompson, K., Czuba, K., Duggan, M., Hiyakumoto, L., Hu, N., Huang, J., Ko, J., Lita, L.V., Murtagh, S., Pedro, V., & Svoboda, D. The JAVELIN Question-Answering System at TREC 2002. http://citeseer.nj.nec.com/nyberg02javelin.html Available online (6 February 2004).

[14] Arguello, J., Elsas, J.L., Yoo, C., Callan, J., & Carbonell, J.G. Document and Query Expansion Models for Blog Distillation. *In Proceeding of the Seventeenth Text Retrieval Conference (TREC 2008)*, 2008.

[15] Ong, S. K., & Tang, E.K. Conceptual Modeling and Reasoning Using Ontologies. *Proceedings. National Computer Science Postgraduate Colloquium 2005 (NaCSPC '05)*, 2005, pp. 27-28.

[16] Shapiro, S.C. A Logic of Arbitrary and Indefinite Objects, in: Principles of Knowledge Representation and Reasoning. In: Proceeding of the Ninth International Conference (KR-2004), 2004, pp. 565–575. AAAI Press, Menlo Park.

[17] Dongilli, P., Franconi, E., An Intelligent Query Interface with Natural Language Support, In *Proceeding FLAIRS Conference*, 2006, pp. 658-663.

[18] Sowa, J. F. *Knowledge Representation: Logical, Philosophical and Computational Foundation*. Pacific Grove, USA: Thomson Learnin, 2000.

[19] Rabiah A. K., T. M. T. Sembok, & Halimah B. Z. Computing Key-word Answer to Hypothetical Queries in the Presence of Skolem Clauses Binding. *Journal WSEAS Transaction on Computer*, 2007, 6(3): 514-521.

[20] Bratko, I. Language Processing with Grammar Rules. In *Prolog Programming for Artificial Intelligence*, 2001, pp. 555-580. Great Britain: Addison-Wesley Publishers.

[21] Clocksin, W. F., & Mellish, C.S. *Programming in Prolog*. Heidelberg, Berlin: Springer-Verlag, 2003.

[22] 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. 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*, 2000, pp. 1-5.

[23] Burhans, D.T., Shapiro, S.C.: Defining Answer Classes using Resolution Refutation. Special Issue of the Journal of Applied Logic on Questions and Answers: Theoretical and Applied Perspectives, 279–295, Available online (18 Jan. 2006).

[24] Bashir, A., Kantor, A., Ovesdotter C. A., Ripoche, G., Le, Q., & Atwell, S. Story Comprehension. http://12r.cs.uiuc.edu/~danr/Teaching/CS598-04/Projects/First/Term2.pdf Available online (11 April 2005).

[25] Rabiah A. K., T. M. T. Sembok, & Halimah B. Z. Binding Skolem Clauses in Theorem Prover Resolution for Automated Hypothetical Question Answering. *Proceedings of the 5th **WSEAS** International Conference on Computational Intelligence, Man-Machine Systems and Cybernetic (CIMMACS '06),* November 20-22, Venice, Italy, 2006, pp. 539 - 607.

[26] Dang, C., & Luo, X., Wordnet-based Document Summarization, In Proceeding of the 7th WSEAS International Conference on Applied Computer & Applied Computational Science (ACACOS ’08), April 6-8, 2008, Hangzhou China, China.