A Unified Logical-Linguistic Indexing for Search Engines And Question Answering

Tengku M. T. Sembok and Rabiah Abdul Kadir

***Abstract*—**Conventional information representation models used in the search engines rely on an extensive use of keywords and their frequencies in storing and retrieving information. It is believed that such an approach has reached its upper limit of retrieval effectiveness, and therefore, new approaches should be investigated for the development of future engines which will be more effective. Logical-linguistic model is an alternative to conventional approach where logic and linguistic formalism are used in providing mechanism for computer to understand the contents of the source and deduce answers to questions. The capability of deduction is much depended on the knowledge representation framework used. We propose a unified logical-linguistic model as knowledge representation framework as a basis for indexing of documents as well as deduction capability to provide answers to queries. The approach applies semantic analysis in transforming and normalising information from natural language texts into a declarative knowledge based representation of first order predicate logic. Retrieval of relevant information can then be performed through plausible logical implication and answer to query is carried out using theorem proving technique. This paper elaborates on the model and how it is used in search engine and question answering system as one unified model.

***Keywords*—**Search Engines, Information Retrieval, Question Answering System, Theorem Proving.

I. Introduction

Question answering has engaged human interest for centuries, including the Greek contribution of Socratic questioning in which deep, systematic, and comprehensive questioning seeks to discover the truth or plausibility of things. Users often have specific questions

which they hope or believe a particular resource can answer. The problem, from the computer system's perspective, is cognitive understanding of the contents in the source and finding the desired answer. Most of the search engines, with Google on the top, able to retrieve most likely relevant information based on a query. But not capable of providing answer to a question due to lack of deduction capability of the engines. In order to find the answer, the engine needs to understand the information content and able to do deduction reasoning.

Search engine (SE) is a kind of information retrieval system. Information retrieval system can be defined broadly as the study of how to determine and retrieve from a corpus of stored information the portions which are relevant to particular information needs. Let us assume that there is a store consisting of a large collection of information on some particular topics, or combination of various topics. The information may be stored in a highly structured form or in an unstructured form, depending upon its application. A user of the store, at times, seeks certain information which he may not know to solve a *problem*. He therefore has to express his *information need* as a request for information in one form or another. Thus IR is concerned with the determining and retrieving of information that is relevant to his information need as expressed by his *request* and translated into a *query* which conforms to a specific information retrieval system(IRS) used. An IRS normally stores *surrogates* of the actually *documents* in the system to represent the documents and the *information* stored in them [1].

II. Human information processing model and IRS model

When a person reads documents to seek for information which are relevant to his needs to solve a problem, he is engaging himself in a highly intellectual process: reading documents written in natural language, using his working memory, and accessing his long term memory in order to understand the documents and decide which are relevant and which are not. This cognitive process of determining the degree of relevance of documents can be expressed based on human information-processing model [2] as depicted in Figure_1a and Figure_1b below.

Tengku Mohd T Sembok is with the Computer Science Department, Kulliyyah of ICT, International Islamic University Malaysia, P.O. Box 10, Kuala Lumpur 50728, Malaysia (phone: +60361966419; fax: +60361965179; e-mail: tmts@ iium.edu.my).

Rabiah Abdul Kadir is with Universiti Putera Malaysia, Serdang, Malaysia (e-mail: rabiah@fsktm.upm.edu.my).

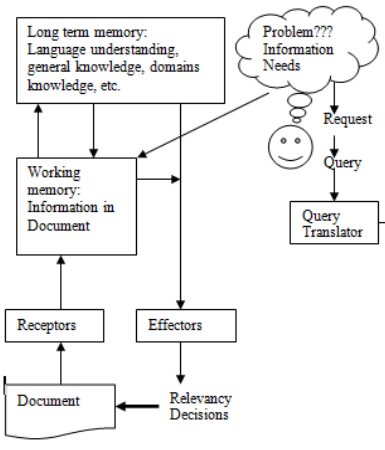

Figure_1a: Human Information-Processing Model

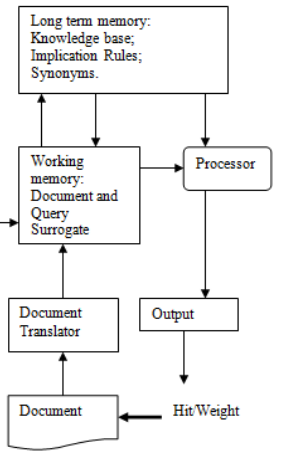

Figure_1b: Information Retrieval System Model

III. Problem Formulation

Most of conventional IRS used bag-of-words approach in indexing and retrieving of information. In these systems, a document is represented by an unstructured collection of keywords or terms which are generally assumed to be statistically independent. The representation does not include any information on syntactic or semantic relationships among those terms. We hold the view that a more accurate representation can be constructed if the method of content analysis takes into account the meaning of information in the documents and query. Thus, a unified logical-linguistic representation is proposed.

The unified logical-linguistic representation is used to index document, retrieve relevant information relating to queries, and provide answers to specific questions. The term unified connotes the used of only one representation to perform these three functions and other functions such as users profiling, synonyms, hyponyms and declaring world knowledge.

Semantic translation process is being used as a major part of the document indexing process. All words in the texts should be identified and given their respective syntactic categories and semantic templates. Document texts are translated into its logical representation which is composed of a set of predicates and logical connectives and transformed to horn clause and variable are Skolomised for ease of theorem proving. Figure 1 depicts the overall process of translation, retrieval and question answering.

IV. Surogates And Representation

In conventional document retrieval systems, the surrogates of documents and queries are built by an unstructured collection of simple descriptors, i.e. the keywords. This representation is not an ideal document or query content indicator for use in IR systems. Given the following titles of documents:

(1) New curriculum and computer facility for management science students,
(2) The undergraduate curriculum in computer science,
(3) 1989 undergraduate computer science curriculum.

It is easy to see that the three independent terms, curriculum, computer and science, characterise all the three titles equally well. While, the phrase computer science is only applicable to titles (2) and (3) only. The representation of a document containing the phrase computer science would be more accurate if the phrase can be derived or established from the document's representation itself. This would allow a query containing the same phrase to fully match with documents like (2) and (3), but not with documents like (1). Going a step further, a good content indicator representation would allow a query with a phrase computer science curriculum to match documents (2) and (3) equally, but not document (1); even though, only document (3) has exactly the same phrase computer science curriculum. In order to do this the retrieval processor, in one way or another, must be provided with enough information to recognise phrases and sentences [3]. In this particular example, a conventional document retrieval system would wrongly match the query containing the phrase computer science curriculum with all the three documents equally well since the information provided by the keyword representation is not informative enough.

The example given above illustrates an obvious shortcoming of the conventional document representation models, such as the vector space model, used in most automatic document retrieval systems or search engines. In these systems, a document is represented by an unstructured collection of keywords or terms which are generally assumed to be statistically independent. The representation does not include any information on syntactic or semantic relationships among those terms. We feel that this kind of representations is too simplified to be highly effective. We hold the view that a more accurate representation can be constructed if the method of content analysis takes into account information about the structure of document and query texts, i.e. the information concerning the syntactic and the semantic structure of the texts. The levels-of-processing theory proposes that there are many ways to process and code information and that knowledge representation used in the memory or storage are qualitatively different.

In order to achieve a more accurate representation of documents and queries, the simple keyword representation ought to be replaced by a knowledge representation such as semantic networks, logic, frame or production system. In our experiment we have chosen logic in the form of first order predicate calculus (FOPC) to represent the contents of documents and queries. A sentence *Mary likes her mother* is expressed in FOPC as the predicate *likes(mary,mother(mary))*.

V. Semantic Representation of Basic English Expression in FOPL

Following the style of Montague Grammar [4][5][6], Table 1 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 programming language.

Table 1: Representation of Simple Words and Phrases

SYNTACTIC CATEGORY	SEMANTIC REPRESENTATION	AS WRITTEN IN PROLOG
CHRISTOPHER (PN)	LOGICAL CONSTANT *christopher*	christopher
ANIMAL (CN)	1-PLACE PREDICATE *(λx)animal(x)*	X^animal(x)
YOUNG (ADJ)	1-PLACE PREDICATE *(λx)young(x)*	X^young(x)
YOUNG ANIMAL (CN with ADJ)	1-place predicate joined by 'and' *(λx)young(x)∧ animal(x)*	X^young(X), animal(X)
WRITES (TV)	2-PLACE PREDICATE *(λy)(λx)writes(x,y)*	Y^X^writes(X,Y)
READ (IV)	1-PLACE PREDICATE *(λx)read(x)*	X^read(X)
is an animal (Copular VP)	1-PLACE PREDICATE *(λx)animal(x)*	X^animal(x)
WITH (PrepP)	1-PLACE PREDICATE *(λy)(λx)with(x,y)*	Y^X^with(X,Y)

The basic expression *animal* and *young*, is a category of CN and ADJ, are translated into predicate (λx)animal(x) and (λx)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 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 (λx)read(x). Meanwhile, a transitive verb such as *writes* translates to a two-place predicate such as (λy)(λx)writes(x,y). The copula (*is*) has no semantic representation. The representation for *is an animal* is the same as for *animal,* (λx)animal(x).

Basic expressions can be combined to form complex expressions through unification process, which can be accomplished by using arguments. The following shows the illustration of combining several predicates in a noun phrase by joining them with ∧ (and) symbol. If young = *(λx)young(x),* smart = *(λx)smart(x),* and animal = *(λx)animal(x),* then, the complex expression will be presented as: young smart animal = *(λx)(young(x)∧ smart(x)∧ animal(x))*. This predicate will be used as index terms *young(x), smart(x),* and *animal(x)* which show their relationship through the argument *x*. Thus, the data structure needed to implement the index for this representation will be more complex than the one implemented for vector space model.

The determiner (DET) can be combined with a common noun (CN) to form a noun phrase. The determiner or quantifier ∃ normally goes with the connective ∧, and ∀ with →. The sentence *An animal called Pooh* contains quantifier and its semantic representation is presented as *(∃x)(animal(x)^called(x,Pooh))*. In this case, Prolog notation is written as `exist(X,animal(X),call(X,Pooh)).`

For this complex expression, the translation is implemented through the unification of arguments in the Prolog's DCG rules. Figure 2 gives an example of English phrase which is translated into FOPL expression illustrated by derivation trees.

VI. QUESTION ANSWERING

Question answering from traditional Artificial Intelligence point of view has been relatively narrowly focused on the task of searching for and returning as answers of an individual that satisfy a query. Consider this current description of question answering, written by Kuhns (1967) in Burhans (2002) [7]:

> *The problem of computerized question-answering is seen to involve a two-step procedure: first, transforming the question into certain sentential formula of the predicate calculus (called the symbolic question); second, generating the answer by calculating (what we shall call) the value set of the transformed question.*
> *The main purpose of the present work is to investigate the second, or answering, process. The process is to output the answer by acting on the symbolic question and the database for the question-answering system. The database consists of a dictionary of description names together with a file of elementary sentences. The value set of a symbolic question containing free variables (e.g., those which stem from natural-language questions such as 'What books has Scott written?') is the list of names which when substituted for the free variables yield a "true" sentence on the database. For a symbolic question without free variables (e.g., one which stems from natural-language questions such as 'Did Scott write Waverly?' the value set, or simply the value, is an expression indicating the truth value.*

Within the purview of AI, 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. Problems of natural language understanding and generation, areas at the heart of AI, come to the fore in question answering. Work on open-domain question answering in large database of documents requires sophisticated linguistic analysis, including discourse understanding and text summarization. The representation of question and answer as well as a reasoning mechanism for question answering are concerns of researchers in knowledge representation and reasoning (KR&R). Formal, mathematical approaches to question answering based on logic and theorem-proving form a subset of KR&R approaches [7].

VII. Previous Work on Question Answering

Some of the early AI systems were question answering systems. The first QA systems were developed as vehicles for natutal 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. BASEBALL answered questions about the US baseball league over a period of one year. LUNAR, in turn, answered questions about the geological analysis of rocks returned by the Apollo moon missions. Both QA systems were very effective in their chosen domains. In fact, LUNAR was demonstrated at a lunar science convention in 1971 and it was able to answer 90% of the questions in its domain posed by people untrained on the system. 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.

Some of the early AI systems included question-answering abilities. Two of the most famous early systems are SHRDLU and ELIZA [8]. 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.

The 1970s and 1980s saw the development of comprehensive theories in computational linguistics, which led to the development of ambitious projects in text comprehension and question answering. One example of such a system was the Unix Consultant (UC), a system that answered questions from the domain of Unix. The system had a comprehensive hand-crafted knowledge base of its domain, and it aimed at phrasing the answer to accommodate various types of users. Another project was LILOG, a text-understanding system that operated on the domain of tourism information in a German city. The systems developed in the UC and LILOG projects never went past the stage of simple demonstrations, but they helped the development of theories on computational linguistics and reasoning.

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. Open-domain question answering is a complex task that needs a formal theory and well-defined evaluation methods [9]. The theory of question answering does not appear in a vacuum. Several theories have been developed earlier in the context of NLP or cognitive sciences. For the task of open-domain question answering against text collection, there are two large-scale end-to-end evaluations: TREC-8 (1999) and TREC-9 (2000).

On the other hand, an increasing number of QA systems include the World Wide Web as one more corpus of text. Currently, there is an increasing interest in the integration of question answering with web search. Ask.com is an early example of such system. Google and Microsoft have started to integrate question-answering facilities in their search engines. In addition, a number of researchers also built systems to take reading comprehension examinations designed to evaluate children's reading levels conducted by Hirschman et al. [10], Charniak et al. [11], Ng et. al.[12], and Riloff & Thelen [13]. These research produced the performance statistics that have been useful for determining how well various techniques chosen work. The same problem has been taken by Bashir et al.[14], on the process of taking short-answer reading comprehension tests. It is still in the interest of question answering research to revitalize research in NLP semantics, such that one can better understand the question, the context in which they are posed, and deliver and justify answers in the context [15].

VIII. Unified Logical-Linguistic Representation

After documents have been retrieved the we presume that the users want to 'talk' to the documents by asking questions. For this reason that the approach of adopted unifies the process of retrieval and question answering. Question Answering (QA) process applies resolution refutation theorem prover as the basic reasoning technique to provide both intentional and extensional answer to question by considering a theorem derived from the question. Since the surrogates for documents are expressed using the unified logic for retrieval purposes, the same surrogates can be used for question answering too. Figure 2 below depicts the architecture of the QA process [16][17][18][19].

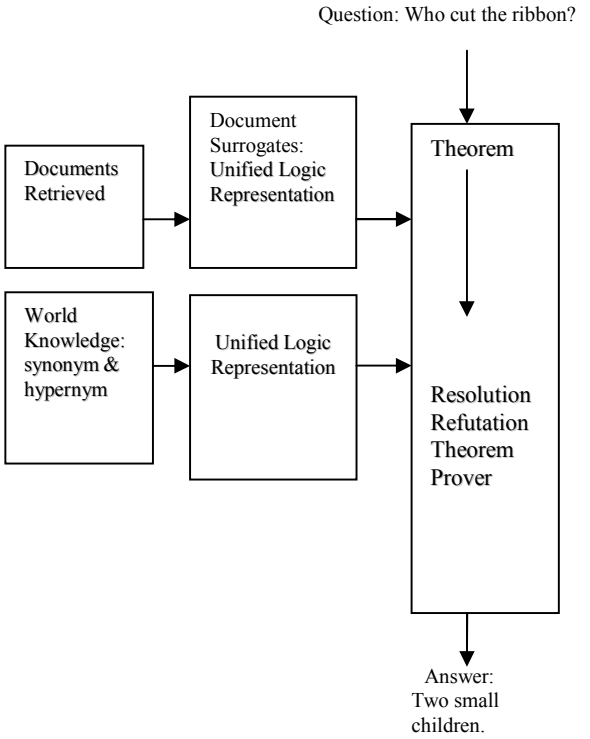

Figure 2: Question Answering Process

IX. Resolution Refutation

Resolution Refutation is an efficient procedure for establishing the truth of a proposition in logical expression. It is a powerful reasoning technique employed in many automated theorem provers. The refutation technique used is that the negation of the theorem to be proven is added to the system, and if it can be resolved to produce the empty clause then a contradiction is established, and the question is answered affirmatively.

For example, if the question asked has the logical form $\exists x\ P(x, y)$, then a refutation proof is initiated by adding the negation clause $\{\neg P(x, y)\}$ to the knowledge base. When the answer literal is employed, the clause $\{\neg P(x, y), ANSWER(y)\}$ is added instead. The argument y in the answer literal (ANSWER(y)) will reflect any substitutions made to it in ¬P(x, y) [19]

Based on example document given in Table 3, if the question given is "Who cut the ribbon?", the negation clause is added to the system would be: Q={ribbon(x), ¬cuts(y,x), answer(y)}. The document would be represented by: KB={…, two(g9), small(g9), children(g9), ribbon(f55), paper(g10), cuts(g9,f55), …} after sklomisation process as shown in Table 4. The resolution refutation proof kept track of the KB to reflect any substitutions made to the variable in Q, we want to show that *(cuts(x,y) ^ ¬cuts(x,y) |— False).* The resolution is considered successful when this is done. In the example this is done when y is substituted by g9 and therefore the answer is "*two(g9) small(g9) children(g9)*".

Table 3: Example of a Document and Questions based on Children Story

World's Tallest Building
The world's tallest building opened today in New York City. It is called the Empire State Building. At noon, two small children cut a ribbon. It was in front of the main door. The ribbon was made from paper. After it was cut, people walked through the door for the first time. Hundreds of people were there. All day long, they took part in a big party on a floor 86 stories high. This building holds as many people as there are in some cities. Each day, 25,000 workers will ride one of the 63 elevators. Another 15,000 people will visit. They might shop or get their hair cut. The Empire State Building is a skyscraper. It is so tall that it seems to scrape the skies. At the very top is a tall, pointed tower. People can go to the top and look at the views. They can see at least 50 miles away.
Question:
1. Who cut the ribbon?
2. When was the ribbon cut?
3. Where is the building?

Table 4: Resolution Refutation Proof Example

Clause	**Substitution**
KB: two(g9)]	
small(g9)	
children(g9)	
ribbon(f55)	
paper(g10)	
cuts(g9,f55)	
makes(f55,g10)	
Q:{ ribbon(f55), ¬cuts(y,f55), answer(g9)	{y/g9, x/f55}
{ answer(g9)}	{two/g9, small/g9, children/g9}

X. Implementation

Indexes of documents are built using the terms in the logical expressions and thus retrieval process is implemented using uncertain logical implication process (see Figure 3) [20][21][22]. The uncertain implication process is used to combine and propagate values that will give a measure of similarity between a document and a query through a process of deduction under uncertainty using their surrogates. In this process each successfully instantiated predicate in the logical representation will be given a value to be combined with other values or propagated to other predicates. Unsuccessfully instantiated predicates are given a zero value. In a logically strict implication process, such as in Prolog, a successfully instantiated predicate is given a TRUE value and an unsuccessfully instantiated one is given a FALSE value. In our case these values are not Boolean, but the real figures based on

statistical calculation, which is the term frequency multiplied by inverse document frequency, i.e. *tf*idf* formulation.

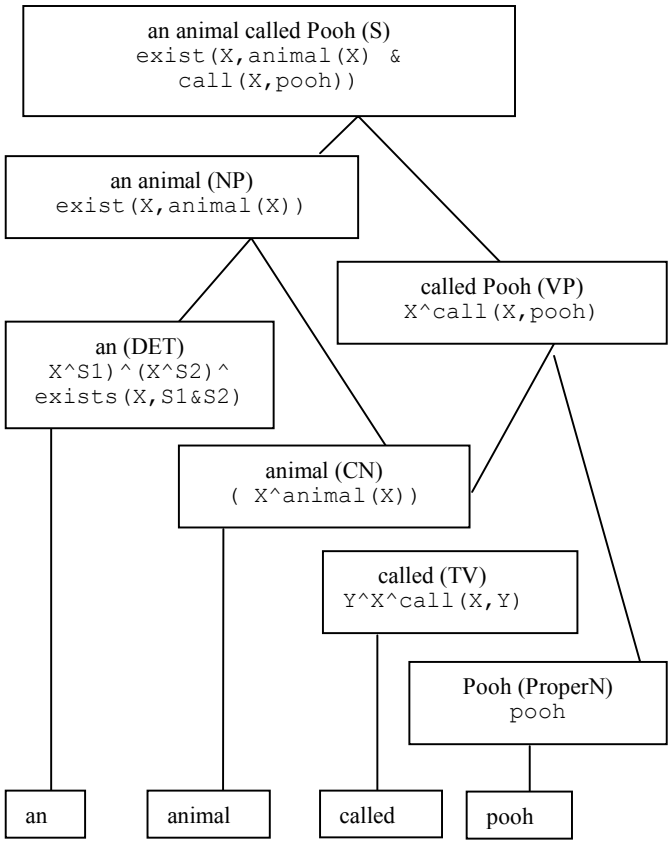

Figure 2: *An animal called Pooh* is translated into logical representation

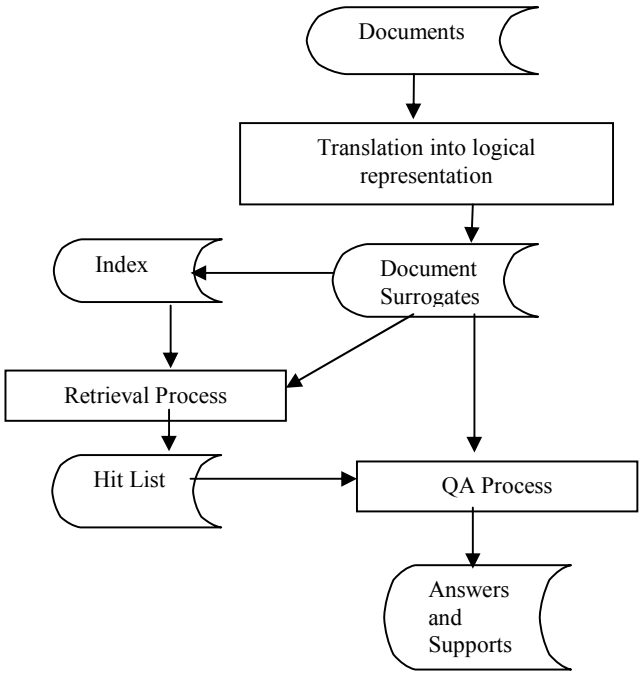

Figure 3: Retrieval and QA Process

An Implication process is used to combine and propagate values that will give a measure of similarity between a query surrogate and a document surrogate. Where should these values come from? It seems that the statistically-based weighting schemes are the best we have so far for this purpose. Thus, in our experiments, the values used are the weights of the stemmed predicate names based on the tf*idf weighting scheme. This weighting scheme is chosen because it is generally considered as being the most effective [21][22]. Thus, the values that will be assigned to the successfully instantiated predicates during this implication process are as follows:

1. For a singleton expression P(x,..y), where P is a stemmed predicate name and x,…y is its argument(s): Its weight, W(P(x)), in a particular document is the tf x idf weight of the word P in the document, i.e. the frequency of P in the document multiplied by its inverse document frequency (idf) value. The idf value of a word of term t is computed using the following formula: idf(t) = - log(Freq(t)/N) where Freq(t) is the number of documents in which the term t appears at least once, and N is the total number of documents in the system.

2. For a complex expression, such as P1(x)&P2(y), Pn(z)&Pi(x,y) the weight is calculated as W(P1(x)+W(P2(y)+..W(Pn(z)+W(Pi(x,y)+(W(P1(x)+W(P2(y) +W(Pi(x,y))/3), i.e. the sum of singleton expressions' weights plus the average weight of the singletons involved in multi-place predicates. Table 2 below illustrates the calculation of the weights.

Table 2: Weight Calculation for Ranking Purpose

Word/Phrase	Expression	Weight
small	small(x)	2
children	children(y)	3
ribbon	ribbon(z)	2
cut	cut(y, z)	3
small Children	small(x)&children(y)	2+3+(2+3/2)=7.5
small children cut a ribbon	small(x)& children(y)&cut(x,z)	3+2+2+(3+2+2)/3 =9.3

The final similarity value between a query and a document is obtained by summing up the values of all predicates in the query surrogate which are successfully instantiated during the implication process. This represents a basis of retrieval strategies applied to our logical model.
This similar idea has been experimented before and has shown a significant improvement over conventional model as far as retrieval effectiveness is concerned, though with slightly different logical expression to represent the surrogates since the QA aspect was not in the scope of experiments [5]. The experiment has used CACM test collection of documents and

queries gathered by Fox at Cornell [23]. This collection contains 3204 "Communications of ACM" articles and 64 natural language queries together with their relevance assessments. Based on precision-recall evaluation the result of the experiment is given in Table 3 below. The benchmark used to evaluate the retrieval effectiveness of the predicate indexing is based on the traditional keywords approach using the *tf x idf* weighting scheme. Table 3 shows the best result obtained using our model as compared to the benchmark based on precision-recall measurement. The figures show an improvement of 24.3% over the benchmark.

Table 3: Recall Cutoff Evaluation Result

	Precisions	
Recall Levels	Benchmark	Our Result
10	52.22	58.74
20	38.52	45.64
30	31.90	38.06
40	24.49	28.64
50	21.01	26.00
60	17.59	22.99
70	12.13	17.68
80	10.23	15.62
90	7.04	11.55
100	6.09	10.14
Average	22.12	27.51
% Increase		24.30

We has also evaluated the system on the performance to answer WH-questions (Who, What, Where, Why, and How) using a data set containing 115 articles with 575 questions and compare the result obtained with human performance [10][11][12][13][14]. Table 4 shows the human performance is better than the system performance by 6%.

Table 4: Comparison with Human Performance in Question Answering

Types of Wh Questions	Performance By: Human	Performance By: System
Who	0.896 (103/115)	0.861 (99/115)
What	0.887 (102/115)	0.861 (99/115)
When	0.922 (106/115)	0.852 (98/115)
Where	0.922 (106/115)	0.930 (107/115
Why	0.809 (93/115)	0.626 (72/115)
Overall Performance	0.887 (510/575)	0.826 (475/575)

Logical representation of documents and queries provides us with a powerful and flexible tool to increase the performance of retrieving relevant documents and answering questions. World knowledge and user profiles can be defined easily to incorporate into the system to guide the retrieval processor in document ranking and provide précised answers to questions. Our next task is to test our idea on a large scale corpus of information. This task will need an efficient data structure formalism and implementation to handle semantic relationship between keywords for storage and deductive processes.

REFERENCES

[1] Mizzaro, S. 1997. Relevance: The Whole History. *JASIS*, Vol.48, No.9, pp.810-832.

[2] Gagne, E., Yekovich, C., Yekovich, F. 1993. *The Cognitive Psychology of School Learning* (2nd Ed) Addison, Wesley, Longman, USA.

[3] Hamzah, M.P., Sembok, T.M.T. 2005. Enhancing Retrieval Effectiveness of Malay Documents by Exploiting Implicit Semantic Relationship Between Words, *Transactions on Enformatika Systems Sciences and Engineering (ENFORMATIKA)*, Volume 10, December 2005

[4] Partee, B. H. (ed.).1976. Montague Grammar. Academic Press, New York.

[5] Sembok, T.M.T., van Rijsbergen, C.J. 1990. SILOL: A simple logical-linguistic document retrieval system, *Information Processing & Management, Vol. 26, No. 1*. Pergamon Press.

[6] Sembok, T.M.T., Zaman, H.B., and Kadir, R.A. 2008. IRQAS: Information Retrieval and Question Answering System Based on A Unified Logical-Linguistic Model. *7th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering And Data Bases (AIKED'08),* University of Cambridge, UK, Feb 20-22, 2008.

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

[8] Martincic, C. J. 2003. QUE: An Expert System Explanation Facility That Answer "Why Not" Types of Questions. *Journal CSC* **19**(1): 336 - 348.

[9] 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. 2002. The JAVELIN Question-Answering System at TREC 2002. http://citeseer.nj.nec.com/nyberg02javelin.html. (Feb., 6, 2004).

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

[11] 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.* 1-5.

[12] 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).* 124-132.

[13] Riloff, E., & Thelen, M. 2000. A Rule-based Question Answering System for Reading Comprehension Tests. *In Proc. of ANLP/NAACL 2000 Workshop on Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems.* 13 - 19.

[14] 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. (April 11, 2005).

[15] Diekema, A. R., Yilmazel, O., Chen, J., Harwell, S., He, L., & Liddy, E.D. 2002. What do You Mean? Finding Answers to Complex Questions. *American Association for Artificial Intelligence*.

[16] Kadir, R.A., Sembok, T.M.T., Halimah, B.Z. 2005. *Question Answering in Reading Comprehension Marking System*. International Conference on ICT in Management (ICTM2005). pp. 933-938. 2005, Melaka, Malaysia.

[17] Kadir, R.A., Sembok, T.M.T., Halimah, B.Z. 2005. *A Logical Deduction to First Order Logic for Use in an Inteeligent Reading*

Comprehension. Proceeding of ROVISP'05. pp 819-823. Penang, Malaysia.

[18] Kadir, R.A., Sembok, T.M.T. and . 2007. Computing key-word answer to hypothetical queries in the presence of skolem clauses binding. *WSEAS Transactions on Computers*;6(3):514-521.

[19] Kadir, R.A., Sembok, T.M.T. and Zaman, H.B. 2009. Improvement of document understanding ability through the notion of answer literal expansion in logical-linguistic approach. *WSEAS Transactions on Information Science and Applications* 2009;6(6):966-975.

[20] Varathan, K.D., Sembok, T.M.T., Kadir, R.A., Omar, N. 2011. Building Knowledge Representation for Multiple Documents Using Semantic Skolem Indexing. *Proceeding of International Conference on Software Engineering and Computer Systems*.

[21] Salton G. 1986. *Recent trends in Automatic Information Retrieval*, Proc. of 1986 ACM Conference on

[22] van Rijsbergen, C.J. 1979. Information Retrieval, 2nd edition, Butterworth.

[23] Fox E.A. 1983. Characterization of Two New Experimental Collections in Computer and Information Science Containing Textual and Bibliographic Concepts, Technical Report 83-561, Department of Computer Science, Cornell University.

Tengku Mohd T. Sembok, the correspondent author, has over thirty years of experience in various fields of Information Communication Technology. He has taught undergraduate and postgraduate programs and managed numerous R&D and consultancy projects successfully. He obtained his B.Sc.(Hons) in Computer Science from Brighton Polytechnic in 1977, MS from Iowa University in 1981, and PhD from Glasgow University in 1989. His previous appointment was as the Deputy Vice Chancellor (Academic and International Affairs) in the National Defence University of Malaysia. He is currently the Dean of College of ICT at the International Islamic University Malaysia.

He has held several academic posts at the National University of Malaysia (UKM) prior to his current assignment. Among the posts held were Head of Computer Science Department, Deputy Dean and Dean of Faculty of Information Science and Technology, and Deputy Director of Research Management Centre. Prior joining UKM he was with Malaysian Rubber Institute and Daresbury Nuclear Physics Laboratory (England) as programmer, and Malaysian Prime Minister's Department as system analyst.

Professor Sembok is a Fellow of Malaysian Academy of Sciences, Fellow of Malaysian Scientific Society and Fellow of British Computer Society. He is the founder and the Chairman of Society of Information Retrieval and Knowledge Management Malaysia.