Logical-Linguistic Semantic in Search Engine

TENGKU M. T. SEMBOK[1]
MOHAMAD FAUZAN NORDIN[2]
ROSLINA OTHMAN[3]
International Islamic University Malaysia
P.O. Box 10, Kuala Lumpur 50728
MALAYSIA
[1]tmts@iium.edu.my
[2]fauzan@iium.edu.my
[3]roslina@iium.edu.my

RABIAH ABDUL KADIR
Universiti Putera Malaysia
MALAYSIA
rabiah@fsktm.upm.edu.my

Abstract: - Users of search engines 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. In order to find a specific answer to a question, the engine needs to understand the information content and able to do deductive reasoning. Conventional information representation models used in the search engines rely on an extensive use of keywords and their frequencies in storing and retrieving information and other characteristic data on specific body of information. It is believed that we need new approaches for the development of future search engines which will be more effective. Semantic model is an alternative to conventional approach. We have proposed logical-linguistic model 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. 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 a 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.

Key-Words: - Search Engines, Information Retrieval, Question Answering System, Theorem Proving.

1 Introduction

Search engine (SE) is a kind of information retrieval system which 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].

2 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 of Gagne et al.[2].

3 Surrogates 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. 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. 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)).*

4 Semantic Representation of Basic English Expression In FOPL

Following the style of Montague Grammar [3][4], 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.

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.

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)

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).

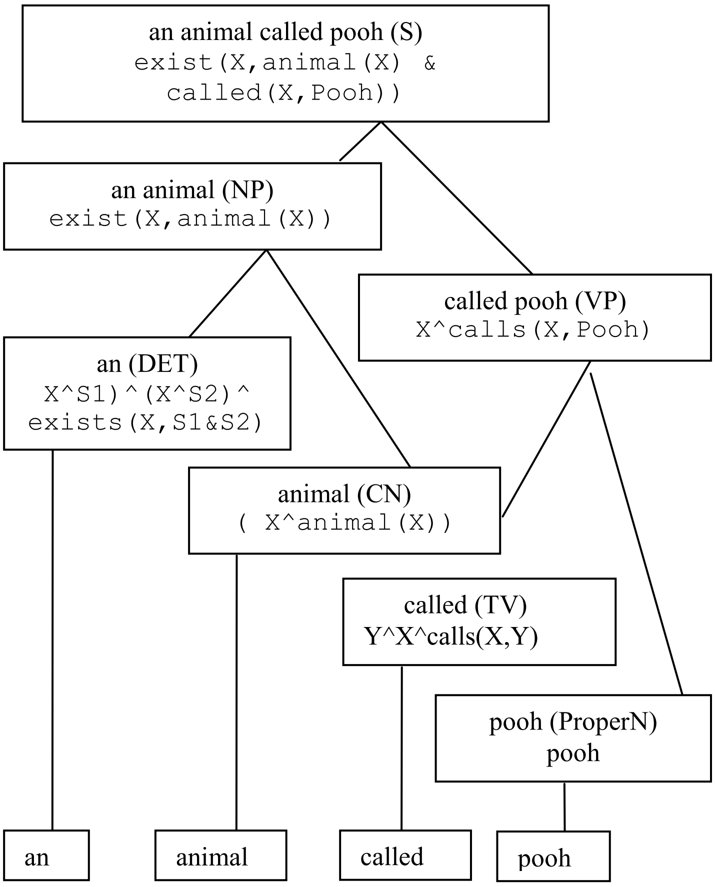
Basic expressions can be combined to form complex expressions through unification process, which can be accomplished by arguments. The following shows the illustration of combining several predicates in a noun phrase by joining them with ∧ (and) symbol. From young = *(λx)young(x),* smart = *(λx)smart(x),*and 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),called(X,Pooh)).
```

Example_1: *An animal called Pooh* is translated into logical representation.



For this complex expression, the translation is implemented through the unification of arguments in the Prolog's DCG rules. Below are examples of English phrases or sentences which are translated into FOPL expressions illustrated by derivation trees.

5 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 1). 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.

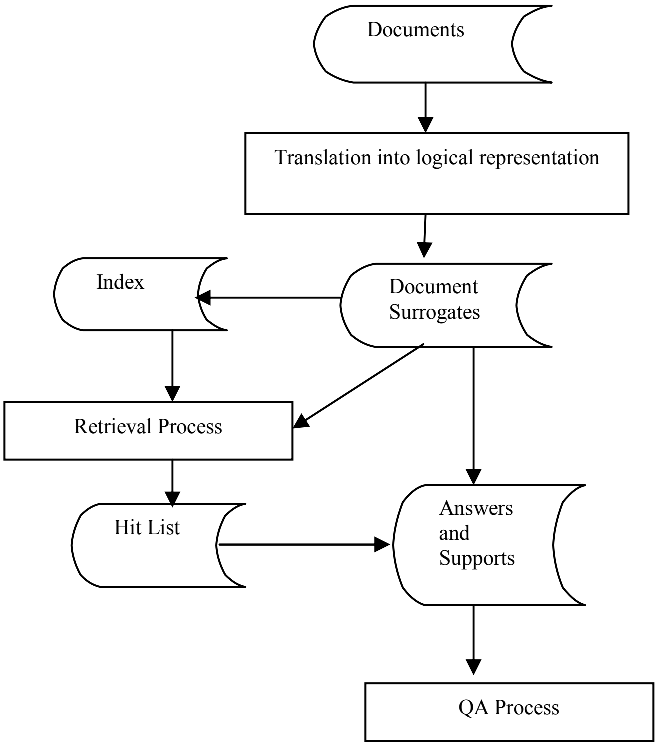


Figure 1: Retrieval and QA Process

World knowledge can be added to help in the implication process by adding rules, such as synonyms and hyponyms rules [5].

Synonym and hyponym rules can be built based on WorldNet database. Similarly, other rules derived from world knowledge which deemed necessary to the implication process may be added to the system. Similar rules can also be added to cater for user profiles such as interest and preferences, e.g. by giving more weight on certain words of interest.

6 Benchmark and Experimental Result

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. Below is the table showing the best result obtained using our model as compared to the benchmark based on precision-recall measurement [6][7]. Table 2 shows an improvement of 24.3% over the benchmark.

We has also evaluated the system on the performance to answer WH-questions using a data set containing 115 articles with 575 questions and compare the result obtained with human performance [8][9]. Table 3 shows the human performance is better than the system performance by just 6%.

Table 2: 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

Table 3: 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)

7 Conclusion

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.

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

[4] 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.

[5] 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*.

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

[7] Salton G. 1986. *Recent trends in Automatic Information Retrieval*, Proc. of 1986 ACM Conference on Research and Development in Information Retrieval, Rabitti F.(Ed.).

[8] 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.*

[9] 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).