Question Answering System using Artificial Intelligence and Fuzzy System

Abhijit Kumar[1], Dr. Lavit Rawtani[2]

[1]Student M.Tech, Department of Computer Science & Engineering
M.A.N.I.T
Bhopal, India
Phone: +919303301885, E–mail: abhijit_mtech@manit.ac.in.

[2] Associate Professor, Department of Information and Technology
M.A.N.I.T
Bhopal, India
E–mail: rawtanilavit@yahoo.co.in.

Abstract— *We will be given sentences from the story, based on this story questions will be asked and the answers to those questions will be given. In this paper we have considered the story of Hare and Tortoise and answered all possible questions based on this story. It is illustrated below with the help of this example.*

Keywords- Predicate logic, Clause form, Resolution

I. INTRODUCTION

We are developing a Question Answering (QA) System for English sentences. The user should be able to access answer of their questions in a user friendly way, that is by questioning the system from the given English paragraph and the system will return the intended answer by searching in context of the paragraph using the repository of English dictionary.

A novel strategy, in addition to conventional search and NLP techniques, will of be used to construct the QA system. The focus is on context based retrieval of information.

To extract passages from the collection of documents an information retrieval engine is needed which can analyze the keywords and passages in detail. The answers to a query the each sentence of the passages is converted into the Predicate Logic using Artificial Intelligence. And then answer is retrieved from the passages using resolution. In predicate logic we can represent real world facts as statements written as well formed formula (wff). It also provides a good way of reasoning with the knowledge.

II. METHODOLOGY

Step 1- To store the whole story into a file.

Step 2- To split each sentences of the story and save it in a different file.

Step 3- Now each sentence is converted into predicate logic and then clause form.

Step 4- Now question is to be asked and converted into predicate logic and then clause form.

Step5- Finally the clause form of the question is resolved to final answer using resolution and unification algorithm..

A. Algorithm: Unify(L1,L2)

1. If $L1$ or $L2$ is a variable or constant, then:
 a) If $L1$ and $L2$ are identical, then return NIL.
 b) Else if $L1$ is a variable, then if $L1$ occurs in $L2$ then return FAIL, else return $\{(L2/L1)\}$.
 c) Else if $L2$ is a variable, then if $L2$ occurs in $L1$ then return FAIL, else return $\{(L1/L2)\}$.
 d) Else return FAIL.
2. If the initial predicate symbols in $L1$ and $L2$ are not identical, then return FAIL.
3. If $L1$ and $L2$ have a different number of arguments, then return FAIL
4. Set *SUBST* to NIL.
5. For $i \leftarrow 1$ to number of arguments in $L1$:
 a) Call Unify with the ith argument of $L1$ and the ith argument of $L2$, putting result in S.
 b) If S = FAIL then return FAIL.
 c) If S is not equal to NIL then:
  i. Apply S to the remainder of both $L1$ and $L2$.
  ii. *SUBST* := APPEND(S, *SUBST*).
6. Return *SUBST*.

B. Algorithm: Convert to Clause Form

1. Eliminate $\rightarrow$, using: $a \rightarrow b = \neg a \vee b$.

2. Reduce the scope of each $\neg$ to a single term, using:

$$\neg (\neg p) = p$$

deMorgan's laws: $\neg(a \wedge b) = \neg a \vee \neg b$

$$\neg(a \vee b) = \neg a \wedge \neg b$$

$\neg \forall x\, P(x) = \exists x \neg P(x)$

$\neg \exists x\, P(x) = \forall x \neg P(x)$

3. Standardize variables.
4. Move all quantifiers to the left of the formula without changing their relative order.
5. Eliminate existential quantifiers by inserting Skolem functions.
6. Drop the prefix.
7. Convert the expression into a conjunction of disjuncts, using associativity and distributivity.
8. Create a separate clause for each conjunct.
9. Standardize apart the variables in the set of clauses generated in step 8, using the fact that: $(\forall x: P(x) \Lambda Q(x)) = \forall x: P(x) \Lambda \forall x: Q(x)$

C. Basis of Resolution and Herbrand's Theorem

Herbrand's Theorem:

To show that a set of clauses *S* is unsatisfiable, it is necessary to consider only interpretations over a particular set, called the *Herbrand universe* of *S*. A set of clauses *S* is unsatisfiable if and only if a finite subset of ground instances (in which all bound variables have had a value substituted for them) of *S* is unsatisifable.

Algorithm: Resolution

1. Convert all the propositions of *F* to clause form.
2. Negate *P* and convert the result to clause form. Add it to the set of clauses obtained in 1.
3. Repeat until either a contradiction is found, no progress can be made, or a predetermined amount of effort has been expended.

a) Select two clauses. Call these the parent clauses.

b) Resolve them together. The resolvent will be the disjunction of all the literals of both parent clauses with appropriate substitutions performed and with the following exception: If there is one pair of literals *T*1 and $\neg$ *T*2 such that one of the parent clauses contains *T*1 and the other contains $\neg$ *T*2 and if *T*1 and *T*2 are unifiable, then neither *T*1 nor $\neg$ *T*2 should appear in the resolvent. If there is more than one pair of complementary literals, only one pair should be omitted from the resolvent.

c) If the resolvent is the empty clause, then a contradiction has been found. If it is not, then add it to the set of clauses available to the procedure.

III. SAMPLE STORY

Story: - Hare and Tortoise ran the race. Hare runs fast. Tortoise runs slowly. Hare took rest for long time. Tortoise runs slowly and continuously. So Tortoise won the race.

A. Conversion of each sentences of story to predicate logic

1. Hare and tortoise ran the race.
Ranrace(Hare , Tortioise)

2. Hare runs fast.
Runs(Hare, fast)

3. Tortoise runs slowly.
Runs(Tortoise, slowly)

4. Hare took rest for long time.
Tookrest(Hare , Longtime)

5. Tortoise runs slowly and continuously.
Runs(Tortoise , slowly) ^ Runs(Tortoise , continuously)

6. Tortoise won the race.
Wonrace(Tortoise)

Some inferred sentences :-

7. $\forall x \forall y$ Ranrace(x,y) ^ Wonrace(x) □ Lostrace(y)

As A □ B => ~ A v B

$\forall x \forall y$ ~[Ranrace(x,y) ^ Wonrace(x)] v Lostrace(y)

$\forall x \forall y$ ~Ranrace(x,y) v ~Wonrace(x) v Lostrace(y)

~Ranrace(x,y) v ~Wonrace(x) v Lostrace(y)

8. Since Long time, Short time etc can be represented better in terms of fuzzy logic as given below :-

Time Period(y)	Membership value μ(y)
Null	0
Very short time	$0 \leq \mu(y) < 0.25$
Short time	$0.25 \leq \mu(y) < 0.50$
Long time	$0.50 \leq \mu(y) < 0.75$
Very long time	$0.75 \leq \mu(y) < 1$
100% duration	1

The values in the membership value and the number of tuples in time period column can vary. As per fuzzy logic the general rule is that if something is in category x of function f_1 then it can not be in category y (Where $y \neq x$) in the same function f_1. This rule can be written as:-

For all x, for all y, for all f_1 if f_1 is a fuzzy logic function and x is a category in f_1 and category is equal to x then category is equal to y.

$\forall x$, $\forall y$, $\forall f_1$, function (f_1) and fuzzy logic set (x, f_1) and fuzzy logic set (y, f_1) and Eq(category , x) => ~ Eq(category , y)

Using fuzzy logic sentences if x is short time then y is not short time. Therefore using the above fuzzy logic equation we get,

Shorttime => ~ Longtime

B. <u>Possible Questions which can be asked</u>

1. Who ran the race?
2. Who runs fast?
3. Who runs slowly?
4. Who took rest for long time?
5. Who took rest?
6. Who runs continuously?
7. Who won the race?
8. Who lost the race?
9. How much time Hare took rest?
10. Did Hare take rest for short time?

C. <u>Conversion of questions to predicate logic then converting it into clause form and then resolving the answer to that question</u>

1. Who ran the race?
   Predicate logic :- Ranrace(x,y)
   Clause form :- Ranrace(x,y)
   Resolution :- ~ Ranrace(x,y) 1
   x = Hare and y = Tortoise
   Answer- Hare and Tortoise.
2. Who runs fast?
   Runs (x, fast) // Predicate logic as well as clause form is same.
   Resolution :- ~ Runs(x, fast) 2
   x=Hare
   Answer- Hare.
3. Who runs slowly?
   Runs(x, slowly)
   Resolution :- ~ Runs(x, slowly) 3
   x=Tortoise
   Answer = Tortoise.
4. Who took rest for long time?
   Tookrest(x, Longtime)
   Resolution :- ~ Tookrest(x, Longtime) 4
   x=Hare
   Answer- Hare.
5. Who took rest?
   Tookrest (x, y)
   Resolution :- ~ Tookrest(x,y) 4
   x=Hare, y= Longtime
   Answer- Hare.
6. Who runs continuously?
   Runs(x, continuously)
   Resolution :- ~ Runs(x, continuously) 5
   x = Tortoise
   Answer- Tortoise.
7. Who won the race?
   Wonrace(x)
   Resolution :- ~ Wonrace(x) 6
   x = Tortoise
   Answer- Tortoise.
8. Who lost the race?
   Lostrace(y)
   Resolution :- ~Lostrace(y) 7
   □ Runrace(x,y) v Wonrace(x) 7
   x =Tortoise , y = Hare
   □ Runrace(Tortoise, Hare) v Wonrace(Tortoise) 6
   □ Runrace(Tortoise, Hare) 1
   As y = Hare has been used so answer => y = Hare
9. Who took rest for long time?
   Tookrest(x,y)
   Resolution :- ~ Tookrest(x,y) 4
   x = Hare , y = Longtime
   Answer- Hare.
10. How much time Hare took rest?
    Tookrest(Hare, y)
    Resolution :- ~ Tookrest(Hare, y) 4
    y = Longtime
    Answer = Longtime.
11. Did Hare took rest for short time?
    Tookrest(Hare, Shorttime)
    Resolution :- ~ Tookrest(Hare, Shorttime) 8
    □ Tookrest(Hare, ~ Longtime) 4
    □ No .
12. Did Tortoise take rest for short time?
    Tookrest(Tookrest , Shorttime)
    Resolution:- ~Tookrest(Tortoise, Shorttime) 8
    □Tookrest(Tortoise, ~ Longtime)
    □No.
13. Did Hare runs continuously?
    Runs(Hare , continuously)
    Resolution :- ~ Runs(Hare, continuously) 5
    □No

IV. CONCLUSION

NLP is a very difficult task because human beings have good common sense and reasoning mechanism which they use to answer the question. Had the task of NLP be easy then

computer would have been told the story, ask questions and computer would have given answers and this preparation of the program would have required very less time.

REFERENCES

1. Agius, H.W., and Angelides, M.C. "Developing knowledge-based intelligent multimedia tutoring systems using semantic content-based modeling", *Artificial Intelligence Review*, 13, 1999, pp 55-83.

2. O'Connor, C., Sceiford, E., Wang, G., and Foucar- Szocki, D. "Departure, Abandonment, and Dropout of E-learning: Dilemma and Solutions", *TechLearn 2003 Conference*.

3. Lempert, R. J., Popper, S. W., Bankes, S. C. "Shaping the next one hundred years: new methods for quantitative, long-term policy analysis, RAND", Santa Monica, CA,2003.

4. Voorhees, E. M., Tice, D. M. "Building a question answering test collection", Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, Athens, Greece,2000.

5. Harabagiu, S.,"FALCON: Boosting Knowledge for Answer Engines". *In Proceedings of the Text Retrieval Conference (TREC-9),2000*.

6. Soubbotin, M., & Soubbotin, S. "Use of patterns for detection of likely answer strings", A systematic approach. *In the Proceeding of TREC 2002*.

7. Roussinov, D., and Robles, J. "Web Question Answering: Technology and Business Applications". *In the proceedings American Conference on Information Systems*. August 6– 8, 2000, New York, NY.

8. Dumais, S., Banko, M., Brill, E., Lin, J., and Ng, A. "Web Question Answering: Is More Always Better?" *Proceedings of the 25th Annual InternationalACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere,Finland, 2002.

9. Yang, H, Chaisorn, L., Zhao, Y., Neo, S., and Chua, T. "VideoQA: Question Answering on News Video".In *Proceedings of the ACM conference on Multimedia (Multimedia'03)*, Berkeley, CA, November 2-8, 2003.

10. Miller, G. "WordNet: An On-line Lexical Database". In *International Journal of Lexicography*, 3-4, 1990.

11. Jawad Makki, Anne-Marie Alquier, and Violaine Prince, "Ontology Population via techniques in Risk Management", ICSWE: Fifth International Conference on Semantic Web Enginnering, Germany,v.1, 2008

12. Farid Meziane, Nikos Athanasakis, Sophia Ananiadon, "Generating Natural Language specifications from UML class diagram" Springer-Verlag London Limited 2007.

13. Pawan Kumar, Rashid Ahmad, Arun Kumar Rathaur, M.K.Sinha, R.Singhal: "Reengineering Machine Translation System through Symbiotic Approach", Accepted in the 3rd International Conference in Contemporary computing, Noida India, August 9-11, 2010. Proceedings to be published by Springer in Communications in Computer and Information Sciences,ISSN: 1865-0929.

14. G.Anthes: Automated Translation of Indian Languages, CACM, Vol.53(1),2010.

15. Artificial Intelligence by Elaine Rich, Kevin Knight & Shivashankar B Nair.

16. Natural Language Understanding by James Allen.

17. Neural network, Fuzzy System and Genetic Algorithm by S. Rajasekharan & G.A.Vijaya Lakshmi Pai.