1.5em 0pt

Interpretation of Natural Language using Data Mining, NLP and Machine Learning Techniques

PROJECT REPORT

submitted by

Anisha Kumari B120535CS

Drishya Praveen C P B120086CS

Indu Sree B120668CS

in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering

under the guidance of

Dr G. Gopakumar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY CALICUT
NIT CAMPUS P.O, CALICUT
KERALA, INDIA 673601
11th MAY 2016

ACKNOWLEDGEMENTS

We would like to express our deep sense of gratitude to the CSE Department for its continuous effort in creating a competitive environment in our college and encouragement throughout this course. We would like to convey our heartfelt thanks to our HOD ***Dr. Abdul Nazeer K A*** for giving us the opportunity to take up this project.

We would like to thank our project guide ***Dr Gopakumar G*** who provided insight and expertise that greatly assisted this project. We would like to acknowledge the tireless and prompt help of our faculty ***Bharat*** Sir. We thank all the staff and non-staff members of the Department of Computer Science and Engineering for helping us directly or indirectly in completing this project successfully. Finally we are thankful to our parents and friends for their continued moral and material support throughout the project work.

ANISHA KUMARI
DRISHYA PRAVEEN C P
INDU SREE

DECLARATION

"We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text".

Place: Calicut
Date: 11-05-2016

Signature :

Name : Anisha Kumari
Roll No.: B120535CS

Signature :

Name : Drishya Praveen C P
Roll No.: B120086CS

Signature :

Name : Indu Sree
Roll No.: B120668CS

CERTIFICATE

*This is to certify that the project report entitled: "***INTERPRETATION OF NATURAL LANGUAGE USING DATA MINING, NLP AND MACHINE LEARNING TECHNIQUES***" submitted by Ms* **Anisha Kumari B120535CS, Drishya Praveen C P B120086CS**, *and* **Indu Sree B120668CS** *to National Institute of Technology Calicut towards partial fulfilment of the requirements for the award of Degree of Bachelor of Technology in Computer Science Engineering is a bonafide record of the work carried out by them under my supervision and guidance.*

Signed by Guide(s) with name(s) and date

Place: Calicut
Date: 11-05-2016

Signature of Head of the Department

Office Seal

Abstract

Our project aims at building a Question Answering System which when posed with the question in natural language, returns an answer motivated by IBM Tool WATSON.Parse tree format of the question is generated using Stanford Parser and from that Triplets are formed using Triplet Extraction algorithm which are then used for retrieving the answer from ontological database.

Contents

1 Introduction **1**
1.1 Problem Statement 1
1.2 Literature Survey 2
1.2.1 WatsonPaths: Scenario-based Question Answering and Inference over Unstructured Information 2
1.2.2 Ontology Based Information Retrieval System for Academic Library 2
1.2.3 Tools and Methods for Building Watson 2
1.2.4 Natural language question answering: the view from here 2

2 Design **4**

3 Implementation **7**

4 Results **12**

5 Conclusions **16**

Bibliography **17**

List of Figures

2.1 Relations 6

3.1 Ontology 8

List of Tables

4.1 Syntax Analysis Results 12
4.2 Semantic Analysis Results 13
4.3 Tone Results 14
4.4 Mood Results 15

Chapter 1

Introduction

A keyword based search engine on being asked a question, presents a list of answers from which it is difficult to figure out the vital information. Question answering system is a form of information retrieval system which gives exact answer to the user's question instead of entire document or paragraphs. IR system works on semantic based reformulation techniques to fish out the accurate answer from the huge number of documents retrieved by the search engine. Watson is an artificial intelligent system that answers question posed in a natural language. It was developed by IBM to answer questions in a quiz show called Jeopardy. It uses advanced natural language processing, information retrieval and machine learning for the task of question answering. It consists of millions of structured and unstructured data in its storage. It parses keywords while searching for related terms in a clue. It makes use of Map-Reduce Algorithm.

It has Applications beyond Jeopardy too, such as :

- Watson can be used to fast-track life-saving research, create connections in unsolved crimes (in seconds).
- Watson for Clinical Trial Matching : Discover new and promising approaches often unavailable outside the setting of clinical trials .
- Watson Curator : Optimize business information to increase user confidence in Watson responses.
- Watson for Oncology : Assistance to oncologists to make more informed treatment decisions.

1.1 Problem Statement

The problem is to develop a miniature prototype of IBM tool Watson using Machine learning Techniques, Language Analysis Algorithms with the help of Apache Jena framework.

1.2 Literature Survey

1.2.1 WatsonPaths: Scenario-based Question Answering and Inference over Unstructured Information

[**1**]WatsonPaths can answer scenario-based questions (Scenario analysis is used to identify information in the natural language accounting the problem scenario useful for solving the problem). For example, when a patient's summary is presented, the answer given is the most likely diagnosis and treatment.

Watson paths are build on the IBM Watson QA system. The input is : natural language questions and output is: precise answers along with accurate confidences. It breaks down the input into smaller pieces of information, asks relevant sub questions to Watson to form new information and represents these results in a graphical model i.e. Assertion graph. Probabilistic inference is performed over the graph to get the final answer and it is repeated till a stopping condition is mentioned.

WatsonPaths takes a two-pronged approach to medical problem solving - First, by expanding the graph forward from the scenario in an attempt to make a diagnosis. Then linking high confidence diagnosis with the hypothesis.

1.2.2 Ontology Based Information Retrieval System for Academic Library

[**2**]The paper deals with the development of a search engine that interprets the meaning of user's query instead of a blind keyword based search. In this paper, an ontology based semantic IR system is proposed which is implemented using Jena semantic web framework and Protege. The input query is parsed by Stanford Parser and sent to triplet extraction algorithm. Later, SPARQL query is formed and it is fired on the knowledge base (ontology) which finds the appropriate RDF Triples in knowledge base and retrieve the relevant information using Jena semantic web Framework.

1.2.3 Tools and Methods for Building Watson

[**3**]This paper discusses the methodology followed by researchers during the development of Watson. The paper describes in detail the software environment, software integration and testing protocol, the DeepQA computing environment built to support huge volumes of high-throughput experiments and the tools they built for system performance measurement and error analysis.

1.2.4 Natural language question answering: the view from here

[**4**]This paper provides an overview of question answering as a research topic in terms of Applications, Users, Question types, Answer types, Evaluation and Presentation. It deals with the main methods of answer construction: through extraction, cutting and pasting snippets from the original documents containing the answer or via generation.

This article discusses about 3 major activities :

- Information retrieval(IR)
- Information extraction(IE)
- Question Answering

IR techniques returns only relevant documents and passages within documents. It provides a methodology for evaluation which resulted in development of QA evaluation. IE is a limited form of question answering in which the questions (templates) are static and the data from which the questions are to be answered are an arbitrarily large dynamic collection of texts.

It explains Generic Architecture for a Question Answering System and the steps involved in it :

- Question Analysis
- Document Collection Preprocessing
- Candidate Document Selection
- Candidate Document Analysis
- Answer Extraction
- Response Generation

Chapter 2

Design

Domain:
A short story : **"The Tiger King"**

Database:
The Story is in the form of unstructured text. Apart from the unstructured text database, there will be two tables maintaining all possible specific and generic questions. Under the generic questions table, we will be maintaining general questions regarding the details of the story and its book like - name, author, name of the publisher etc and its answers. In Specific questions table, we will be writing all possible questions and summary corresponding to each paragraph of the book.

Ontology:

Ontological database for storage of book information using Protege tool is created and this database is used to retrieve related answer from the domain specific ontology.

User interface is created in which user enters input question in natural language.When a question is posed, it is first searched in both Generic and Specific Questions table and if a match is found, then a corresponding answer is returned to the user else proceed with the following steps.

SYNTAX ANALYSIS

Tokenization

Input: Question

Process:
Question is subdivided into tokens using tokeniser which includes lemmatisation using NLTK library by making use of Snowball stemmer. It is written in Python language.

Output: Tokens

POS Tagging

Input: Tokens

Process:
Each token is associated with its part-of-speech tags using NLTK. Some of the POS tags are NN - Noun Singular VB - Verb Base Form NNS - Noun Plural DT -Determiner

Output: Tokens with their corresponding POS tags

Chunking

Input: Tokens with POS tags,Chunk Grammar

Process:
Chunk Parser is formed using Chunk Grammar which segments and labels multi-token sequences. Chunk Parser is used to construct tree structure of question.

Output: Parse Tree

SEMANTIC ANALYSIS

WordNet is a lexical database for the English language. WordNet can help a question answering system to identify synonyms. The synonym information can be used to help match a question with an appropriate rule.

RDF Triple Extraction

Input: Parse Tree

Process :
Triplet Extraction Algorithm is used to extract Subject, Predicate, Object from the tree structure.

1. A sentence (S) is represented by the parser as a tree having three children: a noun phrase (NP), a verbal phrase (VP) and the full stop (.). The root of the tree will be S. Firstly we intend to find the subject of the sentence. In order to find it, we are going to search in the NP subtree. The subject will be

   found by performing breadth first search and selecting the first descendent of NP that is a noun.

2. Secondly, for determining the predicate of the sentence, a search will be performed in the VP subtree. The deepest verb descendent of the verb phrase will give the second element of the triplet.

3. Thirdly, we look for objects. These can be found in three different subtrees, all siblings of the VP subtree containing the predicate. The subtrees are: PP (prepositional phrase), NP and ADJP (adjective phrase). In NP and PP we search for the first noun, while in ADJP we find the first adjective.

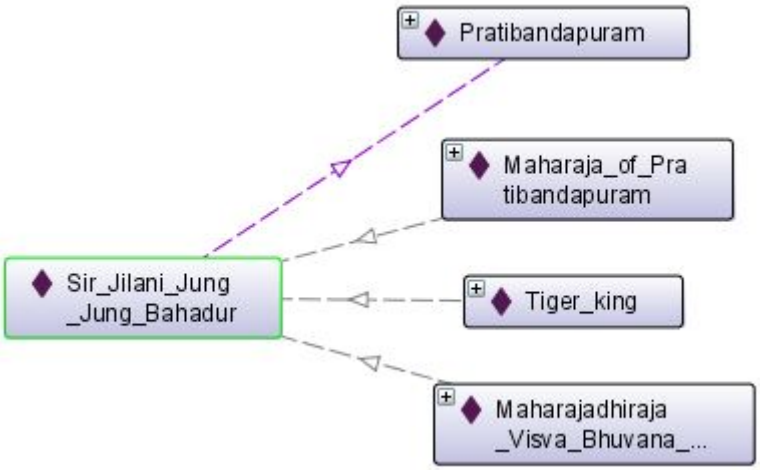

Figure 2.1: Relations

Output: Triple

SPARQL Query Generation

Input: Triple

Process:
Before generating SPARQL query, WordNet or local dictionary can be used to identify synonyms for predicates. If matching predicate is not found in RDF database, Query is generated using Apache Jena framework.

Output: SPARQL query

Information Extraction

Input: SPARQL Query, Ontology

Process:
Jena provides SPARQL API to handle both SPARQL query and their update. Jena API enables the SPARQL query for mapping with RDF database then it is fired on RDF database and retrieves the relevant information performing semantic search into database.

Output: Answer

Chapter 3

Implementation

1. Our project aims at building a prototype capable of answering questions posed in natural language.
2. Ontological database for storage of book information using Protege tool has been created and this database is used to retrieve related answer from the domain specific ontology.
3. User interface has been created in which user enters input question in natural language. When a question is posed, it is first searched in both Generic and Specific Questions table and if a match is found, then a corresponding answer is returned to the user else proceed with Syntax and Semantic Analysis.
4. SYNTAX ANALYSIS
    - Query has been subdivided into tokens using tokeniser which includes lemmatisation using NLTK library by making use of Snowball stemmer.
    - Syntax Analysis phase has been done using Stanford Parser and NLTK. Operations like – Tokenization, Chunking, POS Tagging were performed on the input questions for getting the parse tree.
    - Chunk Parser has been used to construct tree structure of question. It is formed using Chunk Grammar which segments and labels multi-token sequences.
5. SEMANTIC ANALYSIS
    - Triplet Extraction Algorithm has been used to extract Subject, Predicate, Object from the tree structure.
    - RDF Triples have been generated using Triplet extraction Algorithm in the Semantic Analysis phase.
    - Jena Apache Framework has been used to provide SPARQL API to handle both SPARQL query and their update. It enables the SPARQL query for mapping with RDF database, then it is fired on RDF database and retrieves the relevant information performing semantic search into database.
6. DATABASE

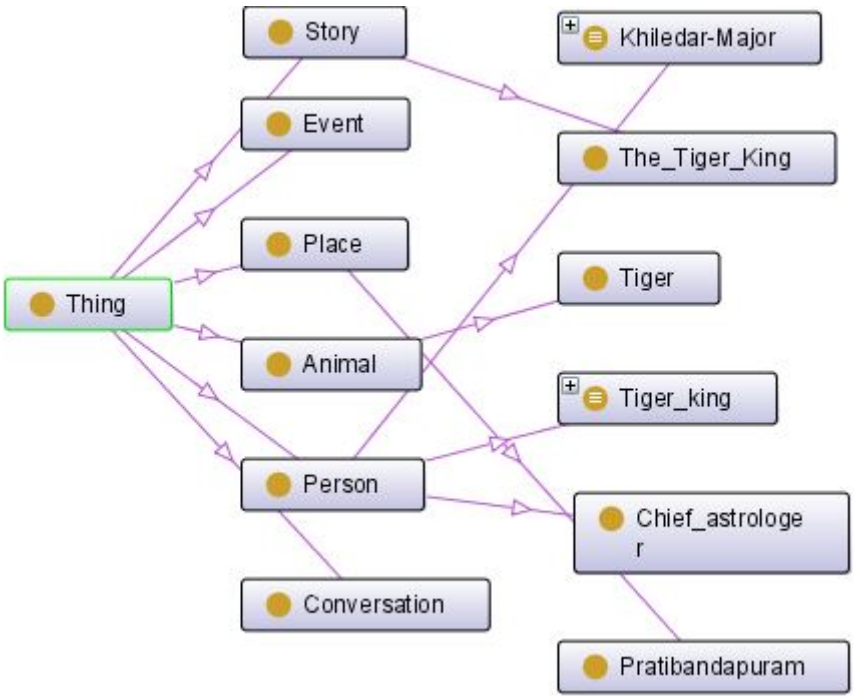

Figure 3.1: Ontology

- A short story "The Tiger King" has been taken. All possible questions from each paragraph with their answers are made and a database is populated. After this a User Interface is created (using HTML, PHP, CSS) where a user can get answers to the direct questions if it is already present in the database.

7. Algorithm to find the mood of a passage has been written and implemented using Bag of Words on random passages. Sample answers are like – Mood is Positive / Negative / Neutral. For implementing this algorithm, separate databases of large number of positive and negative words have been created. When a passage is given as input, the algorithm calculates the number of positive and negative words and displays the mood accordingly.

8. For finding Tone of a passage, a database has been created. It contains many passages with their respective tones which act as a Training set. 5 basic tones have been included in the database namely- Happy, Fear, Motivational, Technical and Aggressive. Using Naive bayes, probability of a random passage is found out and the result is displayed accordingly. The one having maximum probability is the answer.

Here are some of the Algorithms(pseudo codes) which we have implemented :

1. Triplet Extraction Algorithm
2. Finding Tone of a Passage
3. Finding Mood of a Passage

Algorithm 1 TRIPLET EXTRACTION ALGORITHM

```
Input      : Parse tree format of the Question
Output     : Triplets
```

function NP_SUBTREE(sentence)

```
    returns the first subtree with NP as the root
```

function PP_SUBTREE(sentence)

```
    returns the first subtree with PP as the root
```

function VP_SUBTREE(sentence)

```
    returns the first subtree with VP as the root
```

function EXTRACT_OBJECT(sentence)

```
    if the sentence contains (`` '')
       object = all the following words till ('''')
    else
        if VPsubtree is not present
            object = deepest noun of PPsubtree
        else
            Find NP,PP,ADJP,JJR subtrees from VP subtree
                for each value in the subtrees do
                    if value = NP or PP
                        object = first noun in value
                    else
                        object = first adjective in value
    return object
```

function EXTRACT_PREDICATE(sentence)

```
    if VPsubtree is absent
        predicate = deepest noun of NP subtree which is the sibling of PP subtree
    else
        predicate = deepest verb found in VP subtree of the form VB,VBD,VBG etc
    return predicate
```

function EXTRACT_SUBJECT(sentence)

```
    if SBARQ subtree is present
        subject = nouns found in WP or WDT subtrees
    else
        subject = first noun found in NP subtree of the form NNP,NNPS,PRP etc
    return subject
```

Algorithm 2 TONE OF A PASSAGE

```
Input            :   Passage
Training set     :   Files containing passages for each tone(tonefiles)
Selected Tones   :   Happy,Motivation,Fear,Aggressive,Technology
Output           :   Triplets
```

function count_words(tonefile)

```
    returns the number of words in the file
```

function count_aword(tonefile)

```
    returns the number of occurance of a particular word in the file
```

function voc(tonefile)

```
    returns the vocabulary of the file
```

main function

```
    for each tonefile fi do //counting number of words of each tone file
        qi = count_words(fi)
    for each tonefile fi do //counting vocabulary of each tone
        vi = voc(fi)
    for each tone file fi do
        for each word wj in fi do
            p = count_aword(wj,fi)
            sum+=log2((p+1)/(qi+vi)); // Naive Bayes formula using Add one smoothing

        Ci = log2 (1.0/5.0) + sum // Probability of each tone

    M=Max(Cis')

    Return Tone corresponding to M //tone having maximum Ci (probability)

    For passages belonging to more than one tone, a threshold value is set.
        Tones > threshold value --> Selected
        Tones < threshold value --> Rejected
```

Algorithm 3 MOOD OF A PASSAGE

```
Input           :   Passage
Training set    :   Files containing + and - words (positive and negative files)
Output          :   Mood of passage
```

main function

```
    for each word wi of the passage do
        count_pos+=count(wi) in positive file
        count_neg+=count(wi) in negative file

    if( count_pos > count_neg )
        mood = positive

    else if ( count_pos < count_neg )
        mood = negative

    else
        mood = neutral
```

Chapter 4

Results

Syntax Analysis		
Stage	Input	Output
Tokenization	"Who is the king of Pratibandapuram"	['Who', 'is', 'the', 'king', 'of', 'Pratibandapuram', '?']
POS Tagging	['Who', 'is', 'the', 'king', 'of', 'Pratibandapuram', '?']	[('Who', 'WP'), ('is', 'VBZ'), ('the', 'DT'), ('king', 'NN'), ('of', 'IN'), ('Pratibandapuram', 'NNP'), ('?', '.')]
Chunking	[('Who', 'WP'), ('is', 'VBZ'), ('the', 'DT'), ('king', 'NN'), ('of', 'IN'), ('Pratibandapuram', 'NNP'), ('?', '.')]	(ROOT (SBARQ (WHNP (WP Who)) (SQ (VBZ is) (NP (NP (DT the) (NN king)) (PP (IN of) (NP (NNP Pratibandapuram))))) (. ?)))

Table 4.1: Syntax Analysis Results

*URL : http://www.semanticweb.org/dell/ontologies/2016/3/tiger_king

Semantic Analysis		
Stage	Input	Output
RDF Triplet Extraction	(ROOT (SBARQ (WHNP (WP Who)) (SQ (VBZ is) (NP (NP (DT the) (NN king)) (PP (IN of) (NP (NNP Pratibandapuram))))) (. ?)))	SUBJECT-Who Predicate-king Object-Pratibandapuram
SPARQL Query Generation	SUBJECT-Who Predicate-king Object-Pratibandapuram	SELECT ?subject { WHERE ?subject URL king_of URL Pratibandapuram }
Information Extraction	SELECT ?subject WHERE {?subject URL king_of URL Pratibandapuram }	Sir Jilani Jung Jung Bahadur

Table 4.2: Semantic Analysis Results

Identification of Tone	
Passage	Output
True happiness comes from the joy of deeds well done, the zest of creating things new.Our greatest happiness does not depend on the condition of life in which chance has placed us, but is always the result of a good conscience, good health, occupation, and freedom in all just pursuits	HAPPY: -962.084462 TECHNICAL: -1012.975948 FEAR: -960.751470 MOTIVATING: -929.790504 AGGRESSIVE: -977.591932 THRESHOLD: -968 ***MOTIVATING***
As for the boy himself, he was terribly afraid. He could not understand what was happening to him or what he had done. How could he know that his father had taken a hand in killing a daughter of Umuofia? All he knew was that a few men had arrived at their house, conversing with his father in low tones, and at the end he had been taken out and handed over to a stranger. His mother had wept bitterly, but he had been too surprised to weep. IkemefunaâTMs fear stems from deep disorientation, unfamiliarity, and uncertainty about what the future will hold. With a child's limited understanding of the social world around him, he cannot begin to comprehend the series of events that led to his sudden and painful separation from his family. All he knows is that he wants to go home.	HAPPY: -603.726422 TECHNICAL: -634.01372 FEAR: -547.080896 MOTIVATING: -608.141651 AGGRESSIVE: -611.409799 THRESHOLD: -600 ***FEAR***

Table 4.3: Tone Results

Identification of Mood	
Passage	Output
But I feel peaceful. Your success in the ring this morning was, to a small degree, my success. Your future is assured. You will live, secure and safe, Wilbur. Nothing can harm you now. These autumn days will shorten and grow cold. The leaves will shake loose from the trees and fall. Christmas will come, and the snows of winter. You will live to enjoy the beauty of the frozen world, for you mean a great deal to Zuckerman and he will not harm you, ever. Winter will pass, the days will lengthen, the ice will melt in the pasture pond. The song sparrow will return and sing, the frogs will awake, the warm wind will blow again. All these sights and sounds and smells will be yours to enjoy, Wilbur—this lovely world, these precious days. . . "	Mood is positive
And the trees all died. They were orange trees. I don't know why they died, they just died. Something wrong with the soil possibly or maybe the stuff we got from the nursery wasn't the best. We complained about it. So we've got thirty kids there, each kid had his or her own little tree to plant and we've got these thirty dead trees. All these kids looking at these little brown sticks, it was depressing.	Mood is negative

Table 4.4: Mood Results

Chapter 5

Conclusions

The aim of our project is to built a miniature prototype which is capable of answering questions posed in natural language, motivated by an IBM tool called WATSON. Watson has a knowledge base of structured and unstructured data. When posed with a question, it searches for the possible keywords in the query with what is present in its knowledge base. Normal keyword based search engines retrieves results which are optimistic yet they lack in their ability to interpret user question. In this era, a search done by interpreting its meaning is more advantageous than that of blind keyword search.

The project goes through two stages mainly Syntactic Analysis and Semantic Analysis. Syntactic phase deals with Tokenisation, POS tagging and Chunking, which is taken care by Stanford Parser. The output of this phase is a Parse Tree.

The Semantic Phase involves understanding the meaning of the query. It involves Triplet Extraction, SPARQL query generation followed by Information Retrieval. The Triplets (subject-predicate-object) are extracted using Triplet Extraction Algorithm which is fed into SPARQL generation phase and corresponding SPARQL query is formed. This query is then fired upon ontological database which returns the required answer.

As a modification of the project, two distinct features have been added - a.Finding the Mood of a passage b.Finding the Tone of a passage. For mood, sample answers are like – Mood is Positive/Negative/Neutral. For tone, 5 basic tones have been included namely- Happy, Fear, Motivational, Technical and Aggressive. Probability of all the tones of a random passage is found out and the one having maximum probability is the answer.

The project can be enhanced and upgraded by adding a few more features like - Passage Summarization, Finding Moral and Scope, etc. It can also be remodeled by implementing in other languages(other than English). These have not been covered in the project currently because of the time constraint.

Bibliography

[1] Adam Lally, Sugato Bachi, Michael A. Barborak, David W. Buchanan, Jennifer Chu-Carroll, David A. Ferrucci*, Michael R. Glass, Aditya Kalyanpur, Erik T. Mueller, J. William Murdock, Siddharth Patwardhan, John M. Prager, Christopher A. Welty. *WatsonPaths: Scenario-based Question Answering and Inference over Unstructured Information*, RC25489 (WAT1409-048) September 17, 2014.

[2] Amol N. Jamgade and Shivkumar J. Karale, *Ontology Based Information Retrieval System for Academic Library*,IEEE Sponsored 2nd International Conference on Innovations in Information, Embedded and Communication systems (ICIIECS) 2015.

[3] Eric Brown, Eddie Epstein, J. William Murdock, Tong-Haing Fin, *Tools and Methods for Building Watson* , RC25356 (WAT1302-021) February 15, 2013.

[4] L.Hirschman, R.Gaizauskas, *Natural language question answering:the view from here*, 2001 Cambridge University Press.

[5] Haiqing Hu, Peilin Jiang, Fuji Ren and Shingo Kuroiwa, *Web-based Question Answering System for Restricted Domain Based of Integrating Method Using Semantic Information* .