INTERACTIVE CHATBOT

Kavita Pankaj Shirsat
Computer Department .V.I.T
Vidyalankar Institute of Technology
Mumbai,India
kavita.shirsat@vit.edu.in

Dr Satish R Devane
Computer Department.R.A.I.T
Ramrao Adik Institute of Technolgy
Navi Mumbai, India
satish@rait.ac.in

***Abstract*— Intelligent systems are computer programs that aim at providing instruction to humans. In recent years, conversational robots, usually known as chatter bots, become very popular in the Internet, and ALICE (Artificial Linguistic Internet Computer Entity) is probably the most popular one. ALICE brain is written in AIML (Artificial Intelligence Markup Language), an open XML language. They use a huge predefined question Answers database. We have developed a method for answering single answer questions automatically using a collection of sentence. Analysis of texts (both questions and sentences) is based on a statistical part-of speech tagger and phrase recognition. . The system assumes that all the information required to produce an answer exists in a Document provided it.**

***Keywords*—Question Answer (QA) Artificial markup language (A.I.M.L) Natural Language Processing (NLP) Extensible Markup Language (XML) Knowledge Base (KB) Artificial Linguistic Internet Computer Entity (ALICE)**

I. Introduction

Natural language processing (NLP) is a subfield of artificial intelligence and linguistics. It studies the problems of automated generation and understanding of natural human languages. Natural language generation systems convert information from computer databases into normal-sounding human language, and natural language understanding systems convert samples of human language into more formal representations that are easier for computer programs to manipulate. It provides an easy way for human to collect relevant information in more enjoyable way, because chatting has always been fun and if we are able to collect important information and knowledge without any efforts then no doubt it’s a great achievement. Natural language processing is a very attractive method of human-computer interaction. Early systems such as SHRDLU, working in restricted "blocks worlds" with restricted vocabularies, worked extremely well, leading researchers to excessive optimism which was soon lost when the systems were extended to more realistic situations with real-world ambiguity and complexity.

A computer program that implements Natural Language Processing is called chat bot. So a chat bot is to simulate an intelligent conversation with one or more human users via textual methods. Chat bots simply scan for keywords within the input and pull a reply with the most matching keywords or the most similar wording pattern from a local database. Chat bots may also be referred to as talk bots, chatter bots, or chatterboxes. AIML language, which is similar to XML and is based on it, is used to construct brain fot these chat bots. AIML constructs a file and this file stores question sets and answer sets in it. It’s called brain of chatbot because these files serve as knowledge base to the chat bot.

A good understanding of a conversation is required to carry on a meaningful dialog but most chatter bots do not attempt this. Instead they "converse" by recognizing cue words or phrases from the human user, which allows them to use pre-prepared or pre-calculated responses which can move the conversation on in an apparently meaningful way without requiring them to know what they are talking about. For example, if a human types, "I am feeling very worried lately," the chatter bot may be programmed to recognize the phrase "I am" and respond by replacing it with "Why are you" plus a question mark at the end, giving the answer, "Why are you feeling very worried lately?" A similar approach using keywords would be for the program to answer any comment including (Name of celebrity) with "I think they're great, don't you?" Humans, especially those unfamiliar with chatter bot, sometimes find the resulting conversations engaging. The classic early chat bots are ELIZA and PARRY. More recent programs are Racter, Verbots, A.L.I.C.E., and ELIZA. The growth of chat bots as a research field has created an expansion in their purposes. While ELIZA and PARRY were used exclusively to simulate typed conversation, Racter was used to "write" a story called The Policeman's Beard is Half Constructed. ELLA includes a collection of games and functional features to further extend the potential of chatter bot. Chat bots are frequently used to fill chat rooms with spam and advertising, or to entice people into revealing personal information, such as bank account numbers. They are commonly found on Yahoo! Messenger, .NET Messenger Service, AOL Instant Messenger and other instant messaging protocols. Most of the chat-bots are written in A.I.M.L. AIML language, which is similar to XML and is based on it, is used to construct brain fot these chat bots. AIML constructs a file and this file stores question sets and answer sets in it. It's called brain of chat bot because these files serve as knowledge base to the chat bot. AIML is really a powerful language to implement computer interactions with human being. BUT there are some limitations of AIML discussed below:

- Sentence Tokenization
- Pattern Matching
- Morphological Analysis
- Lexical Information
- Syntactic Information

Our Interactive Chat-bot, takes the Document as the input, and save them in the database. Any question asked based on the Document is answered. In this report, we propose few algorithms to explain how our System will work to create the knowledge-base E.L.I.Z.A).In these system different patterns are saved for single sentence, which we avoid in our system. (i.e. the brain of our system) and how it will Answer different types of Question. The main advantage of using such method is that, one doesn't have to save predefined question answer in the database as done in AMIL BASED CHATBOT (e.g. A.L.I.C.E, E.L.I.Z.A).In these system different patterns are saved for single sentence, which we avoid in our system.

And very important that, it has to save huge predefined Question Answer database

II. Proposed System

The architecture of our QA system is displayed in figure I. A user provides an input Sentence and this is processed and saved in the database .whenever the question in asked, Question Analysis is done .Answer Extraction obtains the answer depending on he type of question.

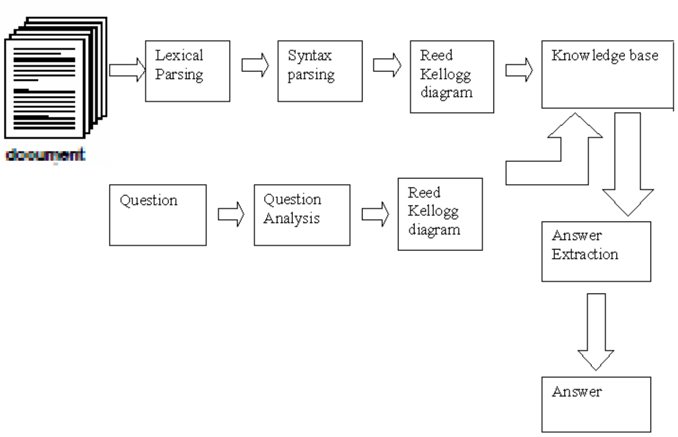


FIGURE I. PROPOSED SYSTEM

A. *Flow Graph of the Proposed System*

The proposed system will work as follows.

1. The system will read the input Document provided to it.
2. Perform Lexical Parsing
3. Perform Syntax Parsing
4. Create a Reed Kellogg Diagram
5. Create a Knowledge –Base(Brain of the system)
6. Read the Question
7. Perform Question Parsing
8. Create a Reed Kellogg Diagram for Input Question
9. The system will apply NLP relationship and accordingly rank the output.
10. The system will display the extracted output.

B. *Natural Language Document Processing*

The Processing of the Document is done as shown in figure II. And the Algorithm for same is written below

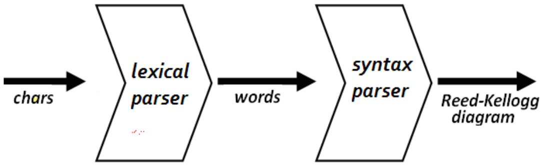


FIGURE II. PROCESSING OF THE DOCUMENT

1) Algorithm for Natural Language document processing

The Algorithm consists of following steps

1. Perform Automatic text summarization and document simplification
2. Perform Lexical Parsing
3. Perform Syntax Parsing
4. Convert it into syntax graphs
5. Build a list of key words for the text
6. Do Spell-checking, based on your own dictionaries

2) Lexical Parsing

Lexical parser is the first one in the pipeline. It takes a string of characters as input and produces Lexemes on output. The parser has a built-in English dictionary, and can be used out of the box. It supports compound words, hyphenation, carrions and more. Application can feed own words when it needs a specific dictionary or when word is unknown for the parser. The parser allows incorporation of non-lexical information in the processed stream like text formatting, DTMF input or arbitrary user data One of the features of lexical parser is word ambiguity. It is useful for integration with OCR or speech recognition when exact word cannot be recognized. Lexical parser produces Lexemes. Lexeme is unbreakable string of characters, it may be a string of white spaces or may have associated Words Word has syntactical information like Part Of Speech and additional syntax tags. This information is used by the second parser in the NLPParser pipeline i.e. the Syntax parser.

3) Syntax Parsing

Syntax parser takes a sequence of Lexemes on input and combines Words into a syntax graph. The graph is a tree of Syntax Nodes. The links in the tree correspond to the syntax roles in a standard Reed-Kellogg diagram like: subject--verb, verb--complement, subject--adjective, verb--adverb and so on. Sometimes links imply a certain sub-tree like Clause, Gerund or Participle. Such structures typically appear on pedestal or tower in classical Reed-Kellogg diagram. Words are leafs in a syntax tree. The graph is called Reed-Kellogg tree graph, because it is essentially a classical Reed-Kellogg diagram but with enforced tree structure. Classical Reed-Kellogg diagram is perfect for human understanding but it's less attractive than tree-based graphs for computer processing. Reed-Kellogg tree representation gives computer program all the advantages of classical Reed-Kellogg diagram combined with simplicity of tree graphs. Reed-Kellogg tree grammar belongs to Type-0 in Chomsky classification and has scalability of human syntax, not reachable in context-free grammars. Sequence of Words linked into a syntax graph gives an Utterance. Usually same words can be joined syntactically in many ways, which results in different meanings. That's why Utterance may be associated with multiple Reed-Kellogg trees. Syntactic ambiguity is an essential feature of NLP for .NET, because a syntax parser producing single syntax graph would be ultimately incorrect. Syntactic ambiguity allows following semantic and pragmatic layers to make a decision about the meaningful interpretation. Utterances are the output of NLPParser and can be used for further semantic analysis.

4) Keyword Extraction

Keyword extraction and analysis is extensively used in information retrieval (in search algorithms) and document clustering (automatic document summarization). Words in an Utterance are syntactically non-equal. Subject, verb or object are more important for understanding a phrase, while adjective or adverb modifiers usually may be omitted or play a helper role by adding more information to the core meaning. This fact may be used when calculating importance of Word for the text. Part of speech also affects Word score. For example, auxiliary verbs can not describe the document theme. Proper nouns are more important than common nouns because they are more unique .Keywords as a characteristic of a theme, do not make much sense for a multi-topic text like books, stories or encyclopedias. Multi-topic texts are typically larger than single-topic texts. Normally they are organized in chapters or articles, which are single-topic. If it is not known in advance whether text is supposed to be mono-topic, the keywords technique may be adjusted. Calculate a list of keywords for a text-window and move this window over a multi-topic text. If set of key-words has changed, it is an indication that topic has changed.

C. *Natural Language Question Processing*

1) Algorithm to Search a Direct Answer for a Natural Language Question

1. For each input question, the program actually checks for a valid question, using Reed-Kellogg syntax function it checks whether we have a proper question.

2. If no question found in syntax tree then it instructs to ask proper question

3. Syntax graphs are matched

4. If syntax nodes match, then meanings of words associated with syntax nodes are compared

5. If both syntax and meanings are equal, and if the utterance are considered to be equal, then matching score is incremented.

6. The more the matches of lexemes, the more the score and the more score gets the output answer.

First, the question (from the upper field) is parsed to get a Reed-Kellogg tree syntax graph. Then the graph is transformed into its direct answer form. For example Question Clause syntax node is replaced with a Clause syntax node. The resulting graph is used as a syntax-lexical pattern

Then the algorithm scans the text in the second field and tries to find utterances, most similar to the pattern. First, it compares a syntax node from the pattern with a syntax node from target text. If syntax nodes match, it compares the meanings of words on the nodes. To compare word meanings it simply compares the Lexemes. If both syntax and meanings match, algorithm goes down the syntax trees and builds the syntax fragment common for both utterances. The more syntax nodes have been matched, the higher is matching score. The best answers are shown as a result.

If question has a question word, the tool assures that question word is always matched. The node in the answer graph, which matches the question word, is the short answer (possibly with all underlying words in a syntax tree)

III. System Implementation

The system is developed with C#.NET. In order to evaluate our proposed method, we have constructed a system that will accept the input statement and will create a database. Further for each type of question that is asked in natural language will answered using specific algorithm so that accurate answer is located.

The required document is saved in the database. The processing of the document consists of Lexical parsing which takes a string of characters as input and produces Lexemes as output. Lexeme is unbreakable string of characters; it may be a string of white spaces or may have associated Words .Word has syntactical information like Part Of Speech and additional syntax tags. This information is used by the second parser in the NLParser pipeline the Syntax parser. Syntax parser takes a sequence of Lexemes on input and combines Words into a syntax graph. The graph is a tree of Syntax Nodes. The graph is called Reed-Kellogg tree graph, because it is essentially a classical Reed-Kellogg diagram but with enforced tree structure. Sequence of Words linked into a syntax graph gives an Utterance. Utterances are the output of NLParser and can be used for further semantic analysis.

Now to search a direct answer for a natural language question. The algorithm takes an input question and converts question form into an answer form. Then algorithm reads utterances from a text file, matches them with the pattern and calculates matching score. Syntax graphs are matched first. If syntax nodes match, then meanings of words associated with syntax nodes are compared. If both syntax and meanings are equal, the syntax node form the pattern and from the utterance are considered to be equal and the matching score is incremented. When all text is parsed, the answers with the best score are suggested. Algorithm makes sure that for questions with a question word the question word is always matched.

In order to evaluate the proposed method we explain our system using appropriate Example. Suppose the system consist of a document with following sentences

Text file.doc

Sachin and Kambli made highest partnership.

They hit 664 runs.

Kambli contributed 349 runs.

Sachin contributed 315 runs.

Kambli was born in Mumbai.

He played last Test match against Lanka.

Who made highest partnership?

Step1: Once the input question is given, the program actually checks for a valid question using Reed-Kellogg syntax function it checks whether we have a proper question.

Question Clause

::subject ['who':0:pronoun]

::verb ['made':2:verb]

::directObject ['partnership':6:noun]

::adjectiveModifier ['highest':4:adjective]

Step2: If no question found in syntax tree then below message is returned If there is no utterances in sentence then it will display "no answer found"

Step3: Then the Syntax graphs are matched the syntax graph for the question who made highest partnership? is as follows

1. Question Clause

::subject ['who':0:pronoun]

::verb ['made':2:verb]

::directObject ['partnership':6:noun]

::adjectiveModifier ['highest':4:adjective]

Step 4: The syntax graph for the matched sentence from the document i.e. Sachin and Kambli made highest partnership? is as follows

Question Clause

::subject ['who':0:pronoun]

::verb ['made':2:verb]

::directObject ['partnership':6:noun]

::adjectiveModifier ['highest':4:adjective]

Step5: If syntax nodes match, then meanings of words associated with syntax nodes are compared here meanings of the words mean "Lexemes" and lexemes mean, if river is noun in questions, and if it is noun in input text then river lexeme is matched.

Step6: If both syntax and meanings are equal, and if the utterance are considered to be equal, then matching score is incremented. The more the matches of lexemes, the more the score and the more score gets the output answer

IV. Results and Analysis

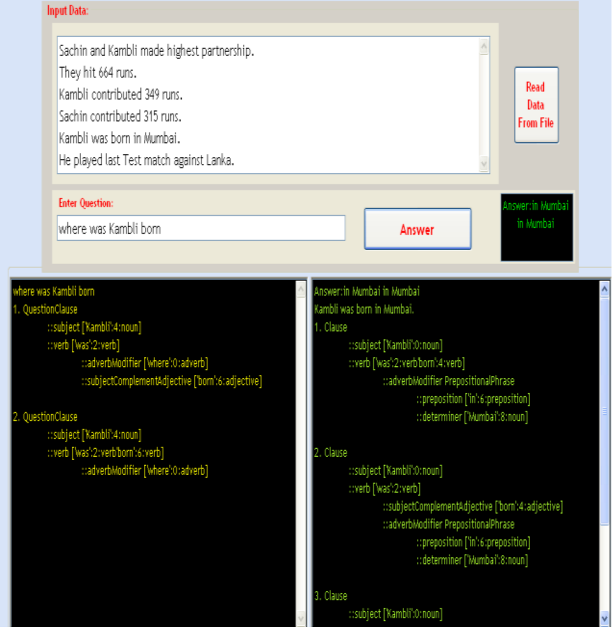
To evaluate our System, an interface was built, which has a button to select the document from the pool of documents based on which question will be asked. It has an input box accept the user question, and a button to send this to the system. The outcomes appear in two columns: one holds the interactive chat answers in one word, and the other is holds the complete Answer along with the match syntax Graph.

A. Results of Implementation

Different documents were selected in order to evaluate our system .Based on the Document Selected Different Types of question were asked by the user to judge the system. The results are displayed in the following sections

1) Result for Direct Question

The Result for the question that has direct answer is as follows: like the question asked was where was Kambli born? And the answer is Mumbai i.e. kambli was born in Mumbai which is the direct answer from the document



```
Sachin and Kambli made highest partnership.
They hit 664 runs.
Kambli contributed 349 runs.
Sachin contributed 315 runs.
Kambli was born in Mumbai.
He played last Test match against Lanka.
```

```
where was Kambli born
```

```
Answer:in Mumbai
in Mumbai
```

```
where was Kambli born
1. QuestionClause
        ::subject ['Kambli':4:noun]
        ::verb ['was':2:verb]
                ::adverbModifier ['where':0:adverb]
                ::subjectComplementAdjective ['born':6:adjective]

2. QuestionClause
        ::subject ['Kambli':4:noun]
        ::verb ['was':2:verb'born':6:verb]
                ::adverbModifier ['where':0:adverb]
```

```
Answer:in Mumbai in Mumbai
Kambli was born in Mumbai.
1. Clause
        ::subject ['Kambli':0:noun]
        ::verb ['was':2:verb'born':4:verb]
                ::adverbModifier PrepositionalPhrase
                        ::preposition ['in':6:preposition]
                        ::determiner ['Mumbai':8:noun]

2. Clause
        ::subject ['Kambli':0:noun]
        ::verb ['was':2:verb]
                ::subjectComplementAdjective ['born':4:adjective]
                ::adverbModifier PrepositionalPhrase
                        ::preposition ['in':6:preposition]
                        ::determiner ['Mumbai':8:noun]

3. Clause
        ::subject ['Kambli':0:noun]
```

B. Empirical Result for different Type Question

The table 8.2 shows the result for different question based on a Single Source Sentence i.e By Using our System we can ask various question by saving only a single sentence in the database and extracting the knowledge from the sentence to answer different Questions were as for same set of question to answer using ALICE CHAT -BOT the brain of Alice requires that many number of AIML sentence to answer the set of Question

TABLE I. EXAMPLES OF MEANINGFUL SENTENCE

Sr. no	**Source Sentences**	**Question**	**Answer**
1	Akbar was a king who ruled India	Who was Akbar	King
		Who ruled India	Akbar
		Which king ruled India	Akbar
		India was ruled by which king	Akbar
		Was Akbar a king	yes
		Was Akbar a queen	no
2	Albert Einstein was a German born Theoretical physicist.	Where was Albert Einstein born?	German
		Was Albert Einstein a physicist	yes
		Was Albert Einstein born in India?	no
		Who was Albert Einstein?	physicist
3	UOS was established in 1918.	When was UOS Established?	1918

C. Comparison of Proposed System with Alice –Bot

Based on Various Factors we compared our System with the Alice Bot And Result of the same is shown in the Table II.

TABLE II. COMPARASION OF PROPOSED SYSTEM WITH ALICE-BOT

Category	**Alice**	**Proposed System**
Database Creation	Manually	Scanned/Copy-Paste Document
language Used	AIML	English
Size of Database	Very Huge	Simple Document
Updating Database	Manually	Scanned/Copy-Paste Document
Time Required for Updating	More	Very Less

Category	Alice	Proposed System
Knowledge-Base	Created in the form of Question Answer	Extracted from the Document
line of code/10 question Asked on Same data	10	1
Answer to direct Questions	90 % Accurate	60 %Accurate
Answer Yes/No Question	40 %Accurate	50 % Accurate
Complex Questions	50 %Accurate	30 % Accurate

D. Analysis of the proposed System

To evaluate the quality of the automatically generated reply, human judges were asked to manually classify each of the 66 human-like replies as, direct Question ,Indirect Question, ,Yes/No Question ,indirect Yes/No Question Based on these classifications, we tested all the algorithm. Figure III gives the evaluation results for the all-in test, which tests the efficiency of the all algorithm given above.

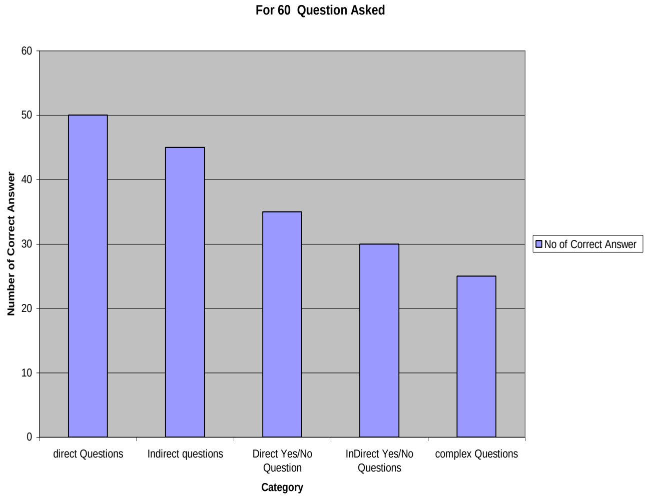


FIGURE III. ALL IN TEST PERFORMANCES

V. Conclusion

Here, we have presented a system which is a new invention in the chatter bot applications .All the chatter bot uses A.I.ML tools and A.I.M.L Language to create the database which is in stored question answer formats. We have designed a system without using A.I.M.L which uses huge number of stored question answer, but were we directly take a Document and store in the database, and based on that data we try to extract the answer to the question using different algorithms.

VI. References

[1] Min-kyoung Kim, Han-joon kim “Design of Question Answering System with Automated Question Generation” *of the* Fourth International Conference on Networked Computing and Advanced Information Management IEEE JULLY 2008

[2] Rolf A Stachowitz “Logics for Knowledge-Based Systems”

International Conference 1988 IEEE

[3] Fernando A. Mikic, Juan C. Burguillo, Daniel A. Rodriguez, Eduardo Rodriguez, and Martín Llamas”T-Bot And Q-BoT Couple of AIML-based Bots for Tutoring Courses and Evaluating Students” International Conference 2008 IEEE October

[4] Calkin A S. Montero and Kenji ARAKI “Information-Demanding Question Answering System” International Symposium on Communications And Information Technologies 2004 (ISClT 2004).

[5] Min-kyoung Kim, Han-joon “Design of Question Answering System with Automated Question Generation” International Conference 2008 IEEE

[6] Giuseppe Altardl and Maria SlML “A Description-Oriented Logic for Building Knowledge Bases” Proceedings of the IEEE, Vol. 74, No. 10, October 1986.

[7] Eric Pacuity Rohit Parikhz Eva Cogan “The Logic of Knowledge Based Obligation” 2005

[8] Rolf A Stachowitz “Logics for Knowledge-Based Systems” International Conference 1998 IEEE.

.