QUIZ GENERATION CHATBOT

A Project Report

Submitted by

Vivek R. Bhave	**111508015**
Shreyansh R. Gopawar	**111508073**
Aditya D. Neralkar	**111508076**

in partial fulfillment for the award of the degree

of

Information Technology

Under the guidance of

Dr. Y. V. Haribhakta

College of Engineering, Pune

DEPARTMENT OF COMPUTER ENGINEERING

AND

INFORMATION TECHNOLOGY,

COLLEGE OF ENGINEERING, PUNE-5

April, 2019

DEPARTMENT OF COMPUTER ENGINEERING

AND

INFORMATION TECHNOLOGY,

COLLEGE OF ENGINEERING, PUNE

CERTIFICATE

Certified that this project, titled "QUIZ GENERATION CHATBOT" has been successfully completed by

Vivek R. Bhave	**111508015**
Shreyansh R. Gopawar	**111508073**
Aditya D. Neralkar	**111508076**

and is approved for the partial fulfillment of the requirements for the degree of "B.Tech. Information Technology".

SIGNATURE
Dr. Y. V. Haribhakta
Project Guide
Department of Computer Engineering and Information Technology, College of Engineering Pune, Shivajinagar, Pune - 5.

SIGNATURE
Dr. V. Z. Attar
Head
Department of Computer Engineering and Information Technology, College of Engineering Pune, Shivajinagar, Pune - 5.

Abstract

Chatbots are proving themselves quite useful now a days. Chatbots like Siri(iOS), Google Assistant(Android), Google Allo (Cross Platform), Cortana (Windows), Amazon Alexa and many more, are used daily by many people, in various parts of world. These chatbots help us accomplish many daily tasks like finding a location, booking a table at restaurant, calling someone, etc.

These chatbots are all open domain chatbots, i.e. they can be used for general purpose tasks. But they are unable to perform specific tasks like answering questions which are specific to industry, education, or an organization. Such chatbots are called as closed domain chatbots. Many companies deploy such chatbots on their website to answer specific questions related to them. Most of the replies are pre-fed and if any new query comes to the chatbot, it either sends the query to the companys concerned authority for a reply or straight away ignores the question.

We will be focusing on how to build an open domain end-to-end conversation model for chatbots which shall test the users on their knowledge on any arbitrary topic they choose. The method used for open domain chatbots can be applied for closed domain chatbots as well if relevant data on that domain is available. The only problem faced by closed domain chatbots is that they lack datasets for development, whereas lots of data is available for open domain chatbots.

Contents

List of Tables

List of Figures

List of Symbols

Chapter 1

Introduction

1.1 What is a Chatbot?

A chatbot is a human-computer dialog system with natural language. A basic framework that outlines the functions expected from modern chatbots can be described as:

ialogic Agent: It must understand the user, i.e. it must provide the function of comprehension. Bots are provided with a textual input, which should be analyzed and generate appropriate responses.

ational Agent: It must have access to an external base of knowledge and common sense i.e huge datasets such that it can provide the function of competence and answering user questions. It should store context-specific information including but not limited to the users personal details, some event the user is currently referring to, etc.

odied Agent: It should provide the function of presence. This was earlier considered to be optional, but today it is considered to be critical for the product to be accepted by the users at large. Today, developers are focused on the use of language tricks to create personas for chatbots in order to build trust with users and give the impression of an embodied agent.

1.2 History of Chatbots

In 1950, Alan Turing asked the question Can machines think?. Turing conceptualized the problem as an imitation game (Turing Test), in which an interrogator asked questions to human and machine subjects, with the goal of identifying the human. If the human and machine are indistinguishable, he said that we could conclude that the machine could think.

In 1966, Joseph Weizenbaum at MIT created the first chatbot that, arguably, came close to imitating a human: ELIZA. Given an input sentence, ELIZA would identify keywords and pattern match those keywords against a set of pre-programmed rules to generate appropriate responses.

Since ELIZA, there has been progress in the development of increasingly intelligent chatbots. In 1972, Kenneth Colby at Stanford created PARRY, a bot that impersonated a paranoid schizophrenic. In 1995, Richard Wallace created A.L.I.C.E, a significantly more complex bot that generated responses by pattern matching inputs against ¡pattern¿ (input) ¡template¿ (output) pairs stored in documents in a knowledge base. These documents were written in Artificial Intelligence Markup Language (AIML), an extension of XML, which is still in use today.

Modern chatbots include: Amazons Echo and Alexa, Apples Siri, and Microsofts Cortana. The architectures and retrieval processes of these bots take advantage of advances in machine learning to provide advanced information retrieval processes, in which responses are generated based on analysis of the results of web searches. Others have adopted generative models to respond; they use Statistical Machine Translation (SMT) techniques to translate input phrases into output responses. Seq2Seq, an SMT algorithm that used recurrent neural networks (RNNs) to encode and decode inputs into responses is

a current best practice.

1.3 Question Answering Chatbot

A Question Answering Chatbot is a kind of chatbot specifically designed only to answer questions posed by the user. The major difference between usual chatbots and a QA chatbot is that the QA chatbot is specifically designed only to answer users questions and not for its other purposes like carrying out commands or performing repetitive tasks. It can be used to answer general questions like those from current events, news articles or those from specific domains only like astrology, astronomy, etc.

1.3.1 Types of QA Chatbot

QA research attempts to deal with a wide range of question types including: fact, list, definition, How, Why, hypothetical, semantically constrained, and cross-lingual questions. There are two type of questioning answering system:

Closed Domain:

Closed Domain system deals with questions under a specific domain like biology or cricket. Closed-domain might refer to a situation where only limited types of questions are accepted. NLP systems that can exploit domain-specific knowledge frequently formalized in ontologies make this task easier.

Open Domain:

Open Domain systems rely on word knowledge and general ontologies and deals with questions about nearly any topic of human interest. Plenty of datasets can be found available for training purposes of such chatbots.

1.4 Architecture & Design

1.4.1 Natural Language Processing

The goal of natural language processing (NLP) is to take the unstructured output and produce a structured representation of the text that contains natural language understanding (NLU). In this section, we explore a number of methods for extracting semantic information from written language in order to create grammatical data structures that can be processed by the Dialogue Management unit in the next step. Likewise, text inputs to a chatbot may contain (i) grammatical mistakes, (ii) disfluencies, (iii) interruptions, and (iv) self-corrections.

Dialogue Act (DA) Recognition One way to extract meaning from natural language is to determine the type of the sentence. The common classification can be made as Assertive, Interrogative, Imperative, Conventional Opening or Exclamatory. Such a classification of sentences is referred to as Dialogue Act Recognition.

In dialogue act recognition systems, a corpus of sentences (training data) is labeled with the function of the sentence, and a statistical machine learning model is built which takes in a sentence and outputs its function. The model uses a number of different features to classify the sentences including but not limited to words and phrases such as please or are you syntactic information like position of subject and object of verb, etc.

Speaker	Dialogue Act	English
A	Conventional-opening	Hallo!?
B	Conventional-opening	Hi Peter!
B	Statement	It's me, Michael.
B	Question	How are you?
A	Conventional-opening	Hello Michael!
A	Statement	Very well.
A	Question	And you?
B	Statement	I'm well too.

Bayesian Approaches to DA Models The idea behind using a Bayesian approach to DA models is to find the probability of every possible sequence of dialogue acts DA that could represent a sentence, and find the dialogue act sequence with the highest probability of occurring.

$$DA_{Max} = argmax_{DA}\, P(DA|U) = argmax_{C}\, \frac{P(DA)P(U|DA)}{P(U)}$$

$$DA_{Max} = argmax_{DA}\, P(DA) * P(U|DA)$$

Assuming the N individual words of the sentence are known, using the Nave Bayes assumptions of independence of subsequent words holds, this leads to:

$$DA_{Max} = argmax_{DA}\, P(DA) * \prod_{i=1..N} P(W_i|DA)$$

This is the unigram model, where P(DA) and P(Wi —DA) can be quantified from empirical data. Using this Nave Bayes classifier, 74% accuracy rate for classifying the correct dialogue act from a given sentence can be obtained. N-gram models are frequently used to include dialogue history in the model. These models estimate P (DA — N historical DAs) rather than P(DA).

Hidden Markov Models can also be used to model dialogue history, where each state represents a dialogue act in the conversation history of the chatbot. Likewise, neural network classifiers can be trained as well. A combination of HMMs and neural networks has achieved 76% accuracy.

Information Extraction The primary responsibility of the NLU is to understand the meaning of the text itself and not just individual phrases. To extract meaning from text, we convert unstructured text into structured grammatical data objects, which will be further processed by the Dialogue Manager. The first step in this process is breaking down a sentence into tokens that represent each of its component parts: words, punctuation marks, numbers, etc. Tokenization is difficult because of the frequency of ambiguous or malformed inputs including: (i) phrases (e.g. New York), (ii) contractions (e.g. arent), abbreviations (e.g. Dr.), and periods (e.g. distinguishing those used in Mr. and at the end of a sentence). These tokens can be analyzed using a number of techniques, described below, to create a number of different data structures that be processed by the dialogue manager.

Bag of Words: Here the sentence structure, order, and syntax is ignored and only the number of occurrences of each word. Stop words are removed from the corpus and lemmatization is carried out. The final bag of words are reduced to a vector space, with each distinct word representing an axes. In the dialogue manager phase, assuming a rule-based bot, these resulting words will be matched against vectors stored in the bots knowledge database to find those with inputs containing similar keywords. The bag of words approach is simple but is not precise enough to solve complex problems.

Latent Semantic Analysis (LSA): This approach is similar to the bag of words. Meanings instead of words, are the basic unit of comparison parsed from a given sentence. Second, groups of words that co-occur frequently are grouped together. In LSA, we create a matrix where each row represents a unique word, each column represents a document, and the value of each cell is the frequency of the word in the document. The distance between the vector representing each utterance and document, using singular value decomposition (reducing the dimensionality of the matrix) is computed and the closest document is determined.

Regular Expressions: Sentences can be treated as regular expressions, and can be pattern matched against the documents in the bots knowledge database. For example, imagine that one of the documents in the bots knowledge database handles the case where the user inputs the phrase: My name is _. where _ is the wildcard character, indicates that this regular expression should be triggered whenever the bot hears the phrase my name is followed by any word.

Part of Speech (POS) Tagging: POS tagging labels each word in the input string with its part of speech. These labels can be rule-based or can also be created using stochastic models which train on sentences labeled with correct POS. POS tagging has plenty of uses in information extraction. In the dialogue manager, POS can be used to store relevant information in the dialogue history. POS is can be used in response generation to indicate the POS object type of the desired response.

Named Entity Recognition(NER): In Named Entity Recognition, the names of people, places, groups, and locations are extracted and labeled accordingly. NER-name pairs can be stored by the dialogue manager in the context history to keep track of the context of the bots current context. Relation extraction goes one step further to identity relations like Who did what to whom and label each word in these phrases accordingly.

Creation of Grammatical Data Structures: Sentences and utterances can be stored in a structured way in grammar formalisms such as context-free grammars (CFGs). Context-free grammars are tree-like data structures that represent sentences as containing noun phrases and verb phrases, each of which contain nouns, verbs, subjects, and other grammatical constructs. Dependency grammars, by contrast, focus on the relationships between words.

1.4.2 Methodology

AIML Chat Bot Implementation AIML chat bot, also known as alice bot is a very simple to use chat bot which could be customized to operate for answering simple FAQ type of questions. It could be changed to reverse the process ,i.e. to create questions given a topic. For this all the documents containing the information about a particular topic is processed to acquire all the sentences on which questions can be generated. The statement can be used to generate questions based on some patterns. It has the following basic tags.

¡aiml¿ : This tag begins and ends the AIML chatbot document

¡Category¿ : This tag marks a "unit of knowledge" in the bots knowledge base.

¡Pattern¿ : This tag is used to contain a simple pattern that matches user input given to an Alicebot. These pattern tags store the context of the conversation and are really useful for a successful chat bot implementation.

¡Template¿ : This tag contains the response to a user inpu

LSTM Implementation Before understanding the LSTM implementation, it is important to understand model of word vectors. One of such word vector model is Word2Vec. Word2Vec is a semantic learning framework that uses a shallow neural network to learn the representations of words/phrases in a particular text. Simply put, it is an algorithm that takes in all the terms (with repetitions) in a particular document (divided into sentences) and outputs a vector form of each.

LSTM are special kind of RNN, capable of learning long term dependencies. For a QA Chatbot, all documents related to particular topic is first passed through word2vec models and their respective embeddings are obtained. The embeddings is then passed through a recurrent neural network and is further added to the existing embeddings. The added information is then passed through an encoder-decoder structure which processes the vectors and creates questions for the particular topic.

Chapter 2

Literature Survey

2.1 Automatic Cloze-Question Generation

This paper describes a system that can generate a list of cloze questions from a document. CQG system is divided into three main module - Sentence selection, Key selection and Distractor selection. In the first stage, informative and relevant sentences are selected from the document. In the second stage, keywords i.e the words/phrases to be questioned on, are identified in the selected sentence. Key selection will not be noun or adjective it would find on the basis of NER. The first two stage are not domain specific. Distractors or answer alternatives for the keyword in the question sentence are chosen in the final stage using web scraping. third and final stage is made domain specific, as quality of distractor depends on domain.

2.2 Automatic question generation on the basis of the discourse connectives

This paper describes a question generation system divided into two modules - Content selection and Question formation. Content selection consists of finding the relevant part in text to frame question form. The Question formation

involves several tasks like Sense disambiguation of the discourse connectives, Identification of question type and Applying syntactic transformations on the content. The authors have concentrated on seven discourse connectives like because, since, although, as a result, for example and for instance. On that basis the Question type will be decided using a rule based model.

2.3 Automatic Question Generation Using Software Agents for Technical Institutions

This paper explains a system which takes as input a text file and the output is another text file containing questions. The system is based on Blooms taxonomy. Blooms Taxonomy is a set of three hierarchical models used to classify educational learning objectives into levels of complexity and specificity. The three lists cover the learning objectives in cognitive, affective and sensory domains. It has been proved that generating questions based on Bloom's taxonomy accurately allows a teacher to judge the learning ability of the students.

The proposed framework helps in question generation by deploying agents, which perform various operations like Document Processing, Information Classification and Question Generation. In Document processing phase stemming is carried out to normalize the senses of the words. Information classification takes an list of keyword generated by Document Processing and finds the Bloom's category of those words, by searching appropriate action verb in the repository which fits with the given keyword. Finally the Question generation module takes the output of Information classification as input to generate questions. The process is a template based approach, which fits the selected keywords in the question template according to the Bloom's levels.

2.4 Automatic Multiple Choice Question Generation System for Semantic Attributes Using String Similarity Measures

The paper describes a system which selects the informative sentence and the keyword to generate a question on, based on the semantic labels and named entities that exist in the sentence. The distractors are chosen based on a similarity measure between sentences in the data set. For generating question Semantic Role Labeler and Named Entity Recognizer is used to identify whether its Name, Location or Name of Organization. Once a question is prepared, then it measures the similarity between the Question sentence and each of the sentences from the Question knowledge. Then the obtained similarity values from other sentences are sorted and three keywords from three different sentences are obtained as distractor values.

2.5 G-Asks: An Intelligent Automatic Question Generation System for Academic Writing Support

This paper describes a system generating specific trigger questions as a form of support for students learning through writing. It describes a large-scale case study, of 24 human supervisors and 33 research students, in an Engineering Research Method course. It compares the questions generated by G-Asks with human generated questions. The paper analyzes the most frequent question types, derived from the human supervisors questions and discusses how the human supervisors generate such questions from the source text.

2.6 Semantic Based Automatic Question Generation

This paper also describes a system that uses both Semantic Role Labeling and Named Entity Recognizer technique to convert the inputted sentence to semantic pattern. It has developed an Artificial immune system which classifies the patterns according to the question type. The question types considered here are the set of WH-questions. Immune system utilizes feature extraction, learning, storage memory, and associative retrieval in order to solve recognition and classification tasks. Inputted sentence will first parse using NER and SRL technique, and from NER and SRL identifies whether, sentences contain person name, location, date. On the basis of this identification, question pattern can be created, i.e if person name then question pattern would be WHO. Sentence pattern for who and question patterns are interpreted as two features vector in the training set, one vector for each question type.

2.7 Automatic Generation Of Multiple Choice Questions From Domain Ontologies

This paper presents an approach that is based on domain specific ontologies and it is independent of lexicons such as WordNet or other linguistic resources. The model developed creates multiple choice question items using the Semantic Web standard technology - Ontology Web Language. The proposed approach is independent of the domain since questions are generated according to specific ontology-based strategies. Class based, property based, terminology based strategies were used to generate the questions. Property-based strategies are described to produce a large number of multiple choice questions but are said to be very difficult to manipulate syntactically. Class

and terminology-based strategies on the other hand are much easier to handle syntactically but generate fewer questions for ontologies of the same depth and population.

2.8 Mind the Gap: Learning to Choose Gaps for Question Generation

In this paper the approach taken is that, the problem of generating good questions is divided into two parts: First, the selection of sentences to ask about, and Second, the identification of which part of the resulting sentences the question should address. To achieve the goal of selecting better gap-fill questions, author has broken down the task into multiple stages like Sentence selection, Question construction, and eventually Classification and Scoring. For generating question, features are used i.e. Token count, lexical, syntactic, semantic and NER feature is used to generate the Gaps fill question.

Sr No.	**Algorithm**	**Methodology**	**Type of Question**	**Evaluation of Result**

1.	Cloze question generation	Sentence selection, key selection and distractor selection is domain specific and NER feature is used for key selection	Cloze	Manually Evaluation is done 1. Evaluation of the selected sentence 2. Evaluation of selected keyword 3. Evaluation of selected distractor.
2.	Automatic question generation on the basis of the discourse connectives	Content selection and Question formation	Question generation like Why, when, where, in which	Manually evaluated for semantic and syntactic soundness of question by two evaluator

3.	Automatic Question Generation Using Software Agents for Technical Institutions	Document Processing, Information Classification and Question Generation.	Define, Describes, Give example, long descriptive questions	-
4.	G-Ask	Citation Extraction, Citation Classification. and Generation	Long descriptive questions like Why, when, Does any..	Compared questions generated by the system to those produced by humans. and Citation Classification performance is done through precision and recall.

5.	Automatic Multiple Choice Question Generation System	Extract sentence from Data Set, Prepare Question sentence, Measure the similarity between the question sentence and all sentences in the knowledge base, Return the three sentences that have the highest similarity values, three keywords of three sentences as distractor selection	MCQ	In this research out of nearly 145 parsed sentences, there were 109 considered good according to the keywords that are extracted from them.

6.	Semantic Based Automatic Question Generation	Input sentence, Feature Extraction through SRL, NER, Choose MCS, Test Sentence pattern and Test the Question type pattern	WH-questions like who, when, where, why, and how.	170 sentences are extracted and mapped into 250 patterns using SRL and NER. The 250 patterns are used in training and testing. and Precision, Recall and F-measurement is used for classification of question type. The percentage of truly generated patterns increased 87% which appears to be promising ratio in this problem comparing it to other techniques used in generating questions automatically.

7.	Automatic Generation of Multiple Choice Questions From Domain Ontologies	Ontology-based strategies like class based, property based, terminology based strategies	MCQ (Choose the correct sentence)	The generated questionnaires were evaluated in three dimensions: Pedagogical quality, linguistic/syntactical correctness and number of questions produced.
8.	Mind the Gap: Learning to Choose Gaps for Question Generation	1)Sentence selection, 2)Question construction, 3)Classification/Scoring.	Fill in the blanks question	manually analyze the generated questions and rate the question

Appendix A

Algorithm

Appendix B

Flow Chart for Making Popcorn