"Where should I eat ?": IOS Application using Watson API’s provided by Bluemix

Sanika Sameer Kulkarni

ABSTRACT
Artificial intelligence is an emerging field of study with cognitive computing as its most important field of research. Cognitive computing is made up of features like natural language processing, machine learning and pattern recognition. These features make machines think like humans and make decisions. IBM Bluemix provides Watson API’s that are beneficial for integrating cognitive computing in applications. There is a lot of data available on the internet which is in unstructured format. Watson understands such unstructured data which can then be made useful. Watson is domain specific, based on the domain training it can give results. This project focuses on creating an IOS application that makes use of cognitive computing using Watson services provided by IBM Bluemix. It provides answers to users’ queries for finding appropriate restaurants in Rochester. Natural language processing being one of the most important feature of Watson, it understands users’ queries in natural language and provides the most suitable answer from its information retrieval algorithm. This IOS application will help users in making decisions as to which restaurants they would like to go to, by answering any question a user would like to know about a restaurant.

Keywords
IBM Bluemix, Retrieve and Rank, Natural Language Processing, QA system

1. INTRODUCTION
Everyone relies on the web whenever they want any kind of information. Almost all answers are found on the internet. There are many applications like Siri[13], Cortana[11] that give you the right information at the right time. Google[6] being one of the most used search engine gives appropriate results to users’ queries in seconds. These search engines require users to enter keywords which are then used to find answers from its huge data collection. Although this saves a lot of users’ time in finding answers it makes humans act like machines. Humans make use of natural language that consists of idioms, slangs, abbreviations, sarcasm to communicate their thoughts to other human beings. Search engines lack in understanding natural language.

Watson was developed with the key idea of understanding natural language. When Watson is trained on a particular domain with all possible resources, it provides solutions identical to how a human would think for a particular problem. It handles unstructured data like images, media files and text. It is different from a normal search engine as it tries to understand users’ queries and provides results with confidence levels so that users do not have to look for the perfect answer from relevant answers provided. Watson uses the DeepQA[4] architecture to interpret users’ queries by using different NLP techniques. First the question is analyzed using parsing techniques and named entity recognition. This helps to derive the four important aspects in the question. These four aspects include the part of question that denotes the hint to an answer, Lexical answer types to indicate named entities, type of question and decomposing of questions into nested sub questions.[8] These techniques break down users’ queries to understand user’s intention. Once Watson is trained for a particular domain it remembers the pattern and provides appropriate solutions for new queries using various answer generation techniques along with the percentage of accuracy. Watson has its applications in many fields like healthcare, telecommunications, weather forecasting, virtual assistants and many more.[14]

People like to go to restaurants once in a while to avoid the mundane routine of cooking all the time. But the only question that pops up in our mind is "Where should I eat ?”. We do not want to go to a restaurant where we spend a lot of money or are not happy with the restaurant. Whether we are residents of a city or new to the city we need to know certain things about a restaurant before going to eat there. Deciding on which restaurants we would like to go depends on factors like ambience, cuisine, cost of meal, mode of payment, distance of restaurant from home, hours of operation, etc. These questions can be answered on the internet by search engines like Google[6] or applications like Yelp[15] or Zomato[17]. For example, if the user enters the question "What should I wear at Amaya Indian Cuisine ?", Google[6] gives a set of documents to the user from which the user needs to find an answer as seen in the Figure 1 or the user could use Yelp[15] and search for the restaurant info first and look at its business info to find "attire". For situations like these, if Watson is trained with a proper set of documents it can understand users’ queries in natural language and
provide immediate results to users’ queries with minimum human assistance to get to the solution. The most important feature of IBM Bluemix is providing Watson API’s that can be integrated into smartphones or desktop applications. This makes it very convenient for the user to get answers to their queries using their mobile devices.

Figure 1: Example of Google not able to do NLP.

This project focuses on “restaurants information” domain. Data of restaurants in Rochester is collected from Yelp[15] and Google[6] and fed into Watson for training. Once Watson is trained it can answer users’ queries in natural language like “Can I pay for my meal using my iphone at Amaya Indian Cuisine?”, “Is there bike parking at Tapas 177?”, “How is the ambience at Tapas 177?”, etc. After Watson is trained it is integrated into IOS using IOS Watson sdk[5]. This IOS mobile application makes it very easy for the user to get information about restaurants, that can help them in deciding which restaurants they would like to go. Watson cannot compare restaurants, so if a user asks “Is Amaya Indian Cuisine better than Tapas 177?”, it cannot answer this query. Watson cannot tell the user as to which restaurants they should go, but it helps in making users’ decision. It can only provide answers to queries based on the data provided to it. For example, for a question like “Give me a list of italian restaurants?” or “How far is Amaya Indian Cuisine from my home?”, requires different set of data and training.

Topics in data management class taught about the importance of cognitive computing. It describes the QA tool called Watson developed for NLP and information retrieval. The coursework also included to research more areas in cognitive computing. During the research I learned about intelligent personal assistants. Also the project needed us to work on IBM Watson experience manager. Once Watson was trained with a proper dataset it was able to understand users’ queries. I also had a lot of experience in IOS mobile development, so wanted to integrate Watson with IOS to make an application similar to intelligent personal assistants. This motivated me to go a step further in using Watson API’s that IBM Bluemix provides for restaurant domain and creating an IOS application as users rely on mobile devices for most of their work.

This paper has six sections. The second section describes some related work in the field of NLP and cognitive computing. It describes some research work done that would help in completing this project. It also describes the advantages and disadvantages of the work done, and how the disadvantages are overcome in this project. The third section describes the architecture and steps taken to complete this project. The fourth section talks about issues faced in implementing this project and the outcome of the project. The fifth section talks about any changes that could be made in the project or any new feature that could be added to the project and also what is learned from this project.

2. BACKGROUND

There is a lot of research work being done to make applications that understand natural language. AnswerBus[16] is one such question answering system that tries to understand natural language. This question answering system accepts users’ queries in natural language form and supports five languages. It answers users’ queries by retrieving information available on the web using search engines. Before retrieving answers from search engines it tries to break down the query to understand the natural language. In the query formation phase it eliminates stop words, and assigns the remaining words into word frequency table. Further for questions that contain “who” marks that sentence to contain a name and for questions that contain “when” marks that sentence to contain a number. Once the query is formed, search engines look for answers on the web by counting the frequency of words in the retrieved sentences. These words need to match the words in the frequency table. These matched sentences are then displayed to the user.[16]
keywords from the keyword dictionary created are extracted. These documents are then parsed through to find the proper sentence that matches to the users’ query. [10]

This paper[7] talks about a question answering system called "START"[7] that uses annotations for NLP. It supports text and media. It contains a huge database that is accessible for retrieving answers. This database consists of data related to a certain topic. For example, a sentence that states "Rain in Mumbai", different annotations related to this sentence are stored in the database. These annotations could include text like "Mumbai has monsoon for four months", "It rains from June to September in Mumbai", "It rains in Mumbai", etc. Also these annotations are retrieved from a document. These annotations are linked to the text "Rain in Mumbai" and linked to documents from which they are obtained. When a user enters a query, "START"[7] tries to match the text in the query with the annotations present in the database. The annotations that match accurately with the query are retrieved and the user is shown the document that contained the annotation.[7]

Figure 2 shows a direct question asked to START[7], "What are the weather conditions in Rochester today?" and Figure 3 shows an indirect question asked regarding the weather condition in Rochester. START[7] fails to understand an indirect question asked in NLP.

Although, these QA systems are able to answer users’ queries, they fail to answer questions when asked indirectly. For example, if you want to know whether a restaurant allows takeout, you could ask a direct question like "Can I order a takeout at Tapas 177?" or an indirect question like "Can I buy my food from Tapas 177 and eat somewhere else?". These QA systems would fail to answer such an indirect question. The reason for such a failure is that, these QA systems look for keywords in the documents which make them act like regular search engines.

3. METHODOLOGY

Watson outperforms most of the question answering systems by understanding natural language. This was proved in "Jeopardy challenge"[3], where Watson won the Jeopardy challenge[3] against humans. Watson can also be implemented into smartphone applications. This is made possible by IBM Bluemix, as it provides Watson APIs that can be integrated into smartphones and desktop applications. This section explains the architecture and implementation of the Watson Question Answer tool as shown in Figure 4.

3.1 Architecture

3.2 Implementation

IBM Bluemix Application Creation:

The first step is to create an account on Bluemix as seen in Figure 5. Then adding services like "Document Conversation"[1] and "Retrieve and rank"[2] from the Bluemix Catalog section. To add these services unique name "documentConv" and "retrieveNRank" is used for Document Conversation[1] and Retrieve and rank[2] tool. These services are bound together in the application. In the retrieve and rank tool a cluster named "restaurant" of size 32gb is created that holds the entire collections database. Inside the cluster a collection named "Recommend" is created which represents a
database in the cloud. Data of restaurants is stored in the collection named "Recommend".

**Data Collection and Preparation**: Data is collected from the business information section in "Yelp"[15]. This data is collected for every restaurant present in Rochester.

Restaurants Information collected includes its price range, hours, address, phone number, attire, ambience, cuisine, parking, wifi, reservations, noise level and takeout. Watson supports documents in .html, .pdf and .docx format. Data of each restaurant is placed in separate word documents in .docx format. Data has to be cleaned and saved in a particular format. The name of Restaurant had to be stored using the "Title" format, heading of parameters like "phone number" had to be stored using "Heading 1" format and the values for those parameters had to be stored using "normal text" format in word document. The document for each restaurant looks as follows in Figure 6. There are 165 word files created for each restaurant. Question set is created containing questions like "What is the phone number of Amaya Indian Cuisine?". A question set of 23430 questions is created. These questions are stored in .txt format as seen in Figure 7. Question set and dataset of restaurants are fed into "Document Conversion" tool. "Document Conversion" tool converts input data into a single document in Json format. This Json file consists of answer units containing a unique id for each restaurant and tags like "body" for parameters and "text" tag for answers to those parameters. This tool shows if the dataset fed into the system is valid or not in the "Content" tab of Retrieve and rank tool. If there is any problem with the format of input data, the document conversion will fail to create a json file. This json file is useful for the retrieve and ranker tool to perform training.

**Training**: In the "Content" section of Retrieve and Rank tool documents and question set uploaded can be seen. The questions that are similar are merged into clusters like "pay", "miscellaneous", "park", etc. Figure 8 shows the clusters. The "Tasks" section of retrieve and rank tool creates tasks of 50 questions related to a cluster as seen in Figure 10. On clicking in a particular task, questions in those tasks are displayed one by one. These questions are displayed with four possible set of answers that are obtained from the retrieve tool. These answers are then rated with stars. One star represents incorrect answer, two stars represent incorrect answer but related to the topic, three stars represent partially correct answer and four stars represent perfect answer. For example, if the question displayed is "What is the address of Amaya Indian Cuisine?", then the answers displayed where "Info - Indian", "Address - Clinton,Rochester", "Price - 11 to 30" and "Parking - Yes". Then the answer "Address - Clinton, Rochester" is given four star rating and rest of the answers have one star rating. Rating can be seen in Figure 11. If the answers displayed does not have a four star rating answer then, "search more answers" tab is selected. In this tab we can enter a keyword that matches the perfect answer, which can later be given four star rating. On obtaining a
During the training, each parameter in the document can be seen with a set of linked question with four star rating answer. This way Watson learns from the set of questions, a natural language query entered by the user. Retrieve and rank tool also gives an option of going back to a particular question in the "Content" tab, to go and change an answer if needed. At any time we can track the number of questions answered and number of tasks remaining to complete the training. Retrieve and rank tool also has an option of adding more documents during training if the question cannot find an answer from the set of input documents given.

The training phase is the most important as the ranker needs to be well trained to understand natural language. "Train a ranker" task is created after every 150 questions. This task trains the Retrieve and Rank tool at periodic intervals, due to which the accuracy of the tool keeps increasing based on the training done. In the "Performance" tab the accuracy of ranker can be seen. "Performance" tab also shows the accuracy of previous rankers which helps in keeping track of the accuracy.

Testing: For better accuracy the ranker should be trained at periodic intervals. As the ranker is trained every time, a "Test" task is created. This "Test" task displays questions and suggests answers, which shows if the ranker can perform accurately. If the ranker suggests wrong answers, we can provide the ranker with perfect answers. This acts as a feedback to the Watson tool and helps it perform better. The "Try out Watson" tab has the option to select a ranker. Based on the ranker accuracy Watson suggests answers for the queries users ask in natural language. Some of the sample questions asked are "What kind of ambience does Tapas 177 have?", "Can I book a table in advance at Amaya Indian Cuisine?", etc. These answers are linked to the documents from which answers are retrieved. There is also an option of comparing the answers suggested by Watson based on different rankers. Figure 12 shows the testing.

IOS Integration: On completion of testing phase and
achieving good level of accuracy, Retrieve and Rank Watson API is integrated into IOS. Watson IOS sdk is integrated into IOS. Parameters required from the Retrieve and Rank tool are “clusterID”, “collection-name” and “rankerID”. UI is created for users to enter their queries. On hitting, “Ask?” button, Watson retrieve and rank api is called that retrieves answers from the ranker. Answers are then displayed to the user on their IOS UI. Figure 13 shows the IOS application.

![Figure 13: Screenshot of IOS application created.](image)

4. RESULTS AND DISCUSSION

4.1 Issues

There were different kinds of issues faced during different phases in the implementation of this project.

**Data Preparation**: In this phase, initially Document conversion tool was displaying the entire document for any queries the user asked. To solve this problem each section in the word document was divided into “Title”, “Header 1” and “normal text”.

**Training**: In the training phase, when the answers displayed did not have a four star rating for a question, answers had to be searched using “search answer” tab. But, sometimes perfect answers were not found in the search answer tab. This issue was solved using document name along with the parameter text in the search answer tab. For example, if the question displayed was “Can I bring my car to Ox and Stone?” and the recommended answers did not have the perfect answer, then in the search tab section “Ox and Stone Parking” was used as keywords to find the correct answer. There was an issue of “Training data has unexpected data” during the “Train a ranker” task. The reason for this issue was, if the question is marked as “I can’t rate this question”, then in the json format of training data “0” value was passed for the question. Watson required at least one answer linked with a question, so the value 0 caused the error. There was an error of “Training data does not meet the requirements”, this error was caused as there was some text in the parameters that had hyperlinks when just text in “h1” format was expected.

**Testing**: “Train a ranker” task was created after every 150 questions. So initially, the ranker was not able to answer natural language questions. The reason for this issue was that Watson needed more number of clusters for every question, so that it understands different ways in which a question could be asked. This issue was solved as more and more questions were trained.

4.2 Final Outcome

With constant training of about 23430 questions and mapping of answers to those questions with perfect four star rating the Retrieve and ranker tool has an NDCG[12] ratio of 0.53 for “miscellaneous” cluster, 0.72 NDCG ratio for “open” cluster, 0.51 NDCG ratio for “pay and eat” cluster and 0.55 NDCG ratio for “pay and meal”. Overall the tool has an NDCG ratio of 0.63 as seen in Figure 14.

![Figure 14: Screenshot of the NDCG ratio of Retrieve and Rank tool.](image)

NDCG ratio shows the possibility of correct answer being retrieved in the top five searched answers. The NDCG ratio lies between 0 and 1, 0 depicts the system performing with very low accuracy and 1 being the perfect performance with maximum probability of suggesting right answers.

Retrieve and Rank tool prefers NDCG over Accuracy. There are some drawbacks when this tool relies on accuracy. The accuracy was calculated by considering the fifth recommended answer. For example, for a question asked, the system recommends answers with first four answers with one star rating (incorrect answer) and the fifth answer with three star rating (partially correct answer), then the accuracy of the system is 50 percent as the score at fifth recommended answer is partially correct. Now for example, if the system recommends answer for a question with first four answers with four star rating (correct answer) and the fifth answer with three star rating (partially correct answer), then the accuracy of the system is 50 percent as the score at fifth recommended answer is partially correct. As seen in both examples, the accuracy of system remains 50 percent in spite of system giving poor performance in first example and good performance in second example. To avoid this, NDCG ratio is considered which is calculated by giving a score to each recommended answer which is equivalent to its number of star rating and average is calculated based on all answers.[9]
A Ranker with good ndcg ratio is integrated into IOS. This IOS application makes it easy for any user to ask questions in natural language and get perfect answers in the top five recommended answers.

5. CONCLUSION
An IOS application for restaurant recommendation integrated with Watson API’s was successfully created. This IOS application gives every user the exact information they need at any time they need. With proper internet connection users can ask queries about any restaurant in natural language into the application and the application connects to the Retrieve and Rank trained model to answer user’s question.

![Figure 15: Screenshot of IOS application connected to Retrieve and Ranker tool and answering users’ query in NLP.](image)

I learned from this project that Watson is the most efficient tool for natural language understanding. If the dataset provided is prepared properly and the training required by the Watson tool is done properly then Watson provides the most accurate results. It outperforms most of the NLP tools available. From this project I learned that it is necessary to test the model at periodic intervals. If the model is not tested at intervals, Watson cannot learn properly and will not give accurate results. Watson understands the pattern of questions, by the clusters of questions formed. It is very important to form clusters of question as Watson remembers the pattern of questions and recommends perfect answers for any new question added. Also, when users enter queries in natural language there is a need for question to be asked in complete sentence form. For example, "Where is Amaya Indian Cuisine located?". Incomplete sentences cannot be used like "Amaya Indian Cuisine location?".

This project could be extended to add restaurants beyond the location of Rochester. Also, Data of restaurants could be collected from Google[6] and Zomato[17] to include more data for parameters. Data such as ratings, reviews and tips given could be collected. This would help in creating more clusters of questions in the Retrieve and Rank tool. More number of clusters would cause an increase in the accuracy. The answer retrieved from the Retrieve and Rank tool is in Json format. This Json file could be parsed to display only the relevant answer instead of showing the entire retrieved text. When users enter queries into the IOS application, it will be checked to see if the question exists in the Watson database. If the question does not exist it could be added as a feedback to the Watson application with a four star rating answer. As, Watson is a self learning machine, it could train itself with the new question added. A question recommender button could be added in the application, that would suggest different questions to first time users. This project can also be extended to Android and desktop applications. Google API can be added to the IOS application, if the users would want to know the distance of a particular restaurant from their current location. Speech to text and Text to Speech Watson API can be added to the application so that the user can ask a question to the application using speech rather than entering text using the keyboard. Conversation, a Watson API can also be added to this App, so that it can act as a chatting application between a user and the machine that would act as a virtual assistant.

6. REFERENCES