Andrew Levandowski

Daniel Hernandez

Ehsan Afshinpour

Prof. Konopka

CS 582

5/7/18

**Abstract**

The growing popularity of chatbots are due to their technological advancements and their emergent ability to accommodate large numbers of users with their questions and automation requests. With the complexities and innovation of artificial intelligence (AI) there is a very large learning curve when it comes to deep learning techniques. This paper gives a brief overview of the main implementations of modern AI used in many consumer chatbots. This paper also provides a stepping stone approach of building a simple hybrid chatbot that incorporates a rule based system with a basic machine learning technique known as a Markov Chain. During our testing and research, we found the strengths and weaknesses of our probabilistic model and how a combination of rule-based and training techniques can lead to more impressive chatbots. Research was conducted through articles about different types of chatbots and certain implementations of them. This hybrid method demonstrates a simple and manageable approach to building one’s own chatbot and demonstrates the techniques and strategies used.

**Problem and Setting**

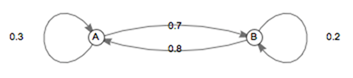
The ability to mimic natural human conversation is extremely complex because of how nuanced language can be. People typically talk to one another to exchange ideas in order to achieve some sort of goal. Technology has tried to model this conversational system to provide assistance in helping humans with remedial tasks. Chatbots emerged as a way to converse with individuals. Although they are characteristically used as advanced command line systems, they have a variety of use and practicality to them. Whether this is for customer service, promotion of advertisement, or just for plain entertainment, chatbots have this inherent nature to provide a realistic experience with the user. With the rise and innovation of artificial intelligence, the resources for programmers and regular consumers alike have provided a multitude of platforms for the creation of these chatbots. However, on the programming side of the creation of chatbots, this requires an in-depth approach and knowledge of machine and deep learning techniques. Many of these topics require months of learning and expertise of these continually updating and advancing subjects. This study provides a basic understanding of chatbots and the implementation of a simple chatbot that incorporates an easier to understand machine learning technique.

For programmers, machine learning techniques can be daunting to learn and understand. The basis of chatbots requires some sort of AI implementation in them in order to cover a wider spectrum of application. Many of these techniques are conceptual and include a lot of formulaic understanding of the underlying mathematics of the machine learning techniques. “Most of the other bots in the market today are rule-based bots because building a truly AI-bot requires building a self-training bot that utilizes natural language processing.” (Shah). Rule-based chatbots are bots that take or look for specific input and have predefined responses. They are easier to comprehend and code because of this predefined simplicity. Although they are limited based on the functionality provided by the programmer, this system of chatbot is a better jumping off point in understanding what it takes to build a chatbot. Although most of the high-end and commercial chatbots use some sort of AI, there is some sort of necessity or want to include some machine learning technique to accommodate the lack there of in a rule-based chatbot. This was the motivation to include what is known to be a Markov Chain with a rule-based chatbot to provide a hybrid platform as a compromise. Many online resources were found to be of either be of a rule-based or AI approach to creating a chatbot. However, this application touches on both techniques.

This study discusses many syntactic and speech processing techniques that were required to make a chatbot, as well as the state machine mechanism of a Markov Chain dictionary. As a brief overview, this bot uses part-of-speech tagging in order to define what the user is trying to say. This incorporates the rule-based system in parsing and analyzing the structure and context of a sentence to construct a meaningful response back. With many AI implementations, general chatbots are built off of big datasets. Datasets, also known as a corpus, are comprised of many lines of text/data. This projects Markoc Chain was built off of Cornell University’s movie corpus. This data set has “220,579 conversational exchanges between 10,292 pairs of movie characters” (“Cornell”). This is used to build a dictionary to mimic the sequences of words that come after one another. This hybrid technique is limited to the sample we used, as in there is only so much we can represent complexities of human speech based on the dataset. This study will show that this technique is limited to semi-coherent responses, but allows for much improvement and learning potential on how to implement more sophisticated AI with a chatbot. The basis of using a dataset and parsing data are paramount for very big and generalized chatbots used in the industry today. This approach also gives insight in how one would construct a rule-based chatbot if needed as well.

**Markov Chains**

Markov chains, named after Andrey Markov, are a mathematical system that experiences transitions from one state to another. Speech recognition, text generation, financial modeling, path recognition and many other AI tools use Markov chains. A Markov chain diagram can have many states and the set of states can be anything such as letters, numbers, weather conditions and so on.



Maltby, Henry. Markov Chains. 14 Jan 2014. Brilliant. Web. 5 may 2018.

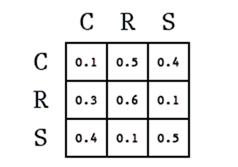
With the above states, we only have two states, but we could have four possible transitions because each state can transition back to its own state. For example, the probability of a process beginning from state B and finishing on A after two moves will be as follows:

Based on the above diagram, in order to start from state B and end on state A after two moves, we either need to stay on state B for the first transition and move to state A as the second move or we need to move to state A and stay on state A as the second move. In this case, the probability will be 0.2\*0.8 + 0.8\*0.3 = 0.4. As you can see, the probabilities of future states are not dependent to the steps that lead to the present state which is called the Markov property. However, not every process has the Markov property.

For another example, let’s say we want to predict the weather for tomorrow and we have two states of sunny and cloudy. So, in this case, we need to have several years of data to calculate the chance of a sunny day after a cloudy day. If the chance of a sunny day after a cloudy day is 0.20, then the chance of a cloudy day after a cloudy day must be 0.80.

**Model**

It is obvious that Markov chain diagrams might seem complicated when there are many states. Therefore, we may use a transition matrix to show the transition probabilities. For N possible states, we will have an N\*N matrix where each cell in the matrix shows us the probability of transitioning from state i to state j. As a result, the cells do the same thing that the arrows do in the previous state diagram. Also, a transition matrix needs to be a stochastic matrix meaning each row shows its own probability distribution and needs to add up to 1.



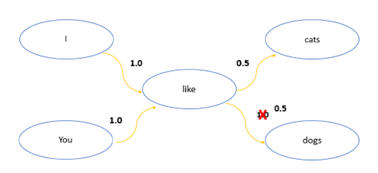
Armstrong, Joseph. Markov Chains-Explained. 8 Jan 2015. Techeffigytutorials. Web.5 may 2018.

Markov chains can become powerful and large. For instance, the algorithm that Google uses for page ranking when it defines the order of each search result, is a kind of Markov chain.

**Markov chains in our Chatbot**

For an example of our Markov Chain, let’s say our first input is “I like dogs”. What our Chatbot needs to do is read these inputs and split up each word into individual states. Then each state or word needs to be connected to the next state so we can transition from one state to another. Also, we need to record the probability of each state. Since this sentence is the only information that we have in our dictionary file, the probability of each transition will be 1.

Now if we add our second input such as “You like cats”, then our diagram and the probabilities will change. The reason is because the word “like” has been used twice as the input. Therefore, the probability of each transition from the word “like” to “dogs” and “cats” will change to 0.5.



**Generating Text**

To generate sentences, we need to pick a state based on the user's input and follow the arrows which are future states. In our example, if the input-based state is the word “You”, then this state only transitions to the state “like” since its probability is 1. Now the word “like” can transition to both states “cats” and “dogs” with the same probability of 0.5 each. Therefore, the Markov chain can pick either one of these two states because they have the same transition probability. Finally, the generated text could be “You like cats” or “You like dogs”.

**Dictionary Format**

We use a dictionary file in our project which is a JSON file. Inside this dictionary file, we store each word followed by the next word and its frequency of occurrence. Using probabilities in the dictionary instead of frequency may make the computation hard; therefore, using the frequency of occurrence is a simpler option. By frequency of occurrence I mean that we keep track of how often each word leads to another word. The following picture shows the way that our JSON file stores the data.



**Contextualizing User Input**

In order to properly respond to a user message, there is a necessity to understand what is being said. There needs to be a way to find the context within a sentence. This was achieved by the aid of TextBlob. “TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.” (“TextBlob”). TextBlob was primary used for its ability to tag parts-of-speech. It contained a function that took in a sentence and marked certain words with ‘NN’ for noun, ‘JJ’ for adjective, ‘PRP’ for personal pronoun, etc. Based on this information we applied rules for the bot to follow, hence our rule based system. For example, if the bot found the pronoun ‘I’, followed by a verb, and also containing a noun in the sentence then a typical response would be constructed with the pronoun ‘You’ followed by the same verb. Then we would insert the noun to search the dictionary to construct the rest of the sentence. The logic behind this was since the user referenced themselves in their input, then naturally the bot would reference the user with the pronoun ‘you’ followed by what the user was talking about given by the noun. Rules such as these give way to how sentences are produced. These rules are based on logical occurrences of what a normal conversation could look like. Based on our interpretation of people speaking to one another, allowed us to place these rule to guide the bot to say something coherent based on what a person says.

**Testing**

During the testing and revising of our chatbot, we learned the strengths and weaknesses of the different types of chatbot techniques. Overall, the two main methods of bot creation may be better suited for different use cases, but are stronger and more flexible when combined. Generally, the training or learning technique that we focused on would thrive in scenarios with larger and more diverse data sets like social media, and a rule-based approach is extremely effective in a more strict and focused environment like online commercial support or user correspondence. As we created sample inputs for our bot in the final stages of our project, it became clear that the amount of training and the quality of a data set are paramount in creating a more intelligible bot. While the Cornell Movie corpus we selected was one of the largest we could find and had somewhat general dialogue, our bot could have benefitted from more natural and diverse conversation subjects. This could also be helped by introducing more data that comes from the same source but reaches a wider variety of subject matters and language formats. This was mainly evident when prompts given to the bot contained words it had not experienced before or when the subject or meaning of the returned response was more suited to a different context. For example, due to the limitations of our movie script data, our bot would occasionally return a non-sequitur to a seemingly common word or use a different meaning of a word like “running out” of something instead of “running” as a form of exercise. While a better set of data and better application of the data could improve issues like these, the most profitable chatbots that are used for online support can benefited further by rule based systems.

**Results**

A support chat on a website for a widely used service is one of the best examples for use of a chatbot as the business has the most to gain in simplifying their user experience. A chatbot in this situation would have a more refined and smaller list of possible subjects and dialogue to be concerned with due to the bot’s specific nature. This partially eliminates one of the highest costs of rule-based systems which is imagining different conversation scenarios and programming the logic behind handling those scenarios. A good example of a mixed rule-based and training system in action is with service providers like Charter Communications. After partnering with Next IT for chatbot services, Charter experienced an 83% lower volume of human to human support chat and a 44% less expensive support solution (Next IT). Chatbots are becoming cheaper and more efficient than human support specialists largely in thanks to the great application of rule-based systems and the introduction of learning techniques into commercial spaces. Rules can simplify and strengthen a bot’s ability to assist a customer while training can make it more adaptable so that it can assist with things its programmers had not specifically thought of. Applying this information back to our bot shows that a combination of both techniques is very beneficial. Even though having a better corpus and more advanced interpretation of training data and test input would make our bot more coherent, a better implementation of our mixed technique idea would be more applicable to real world use, whether that’s commercial, research or other something else. After the completion of our implementation, it became clear that our approach has been visited by others previously and likely already exists in some systems to some degree. After a comparison between different methods, it was found that “… a chatbot which employs both statistical NLP methods as well as heuristic methods, such as pattern matching rules, functions more realistically than a chatbot which only uses one approach or the other” (Chantarotwong 10). For the purposes of our demonstration and learning about something less familiar to us, we decided to focus on the probabilistic Markov Chain model but our hypothesis was correct. Given more time to add to our chatbot, we would want to either add further training capabilities to make a more coherent bot, or add additional and more complex rules to our bot to add to its functionality and its relevance to our original hypothesis.

**Conclusion**

A Markov chain is a simple concept and a stochastic model which represents a sequence of possible events such as a sequence of values or words in a sentence. Speech recognition, text generation, and many other machine learning tools use Markov chains. We used this technique in our project; however, there are always many ways to create and improve this project. During our testing we found some areas of improvement in our bot’s probabilistic technique and data handling and how more reliance on rules would be a more practical solution. This also showed that our idea that a combination of the two techniques was valuable and that similar mixed technique chatbot systems have been envisioned or created. Ideas brought to our attention during our presentation and demonstration were focused on our approach to the probabilistic model technique. When analyzing the frequencies of our data set words, we relied on straight statistics instead of log statistics that would lessen the skew of our data. This decision had less of an effect on our project likely due to the breadth of our data. Any skew that existed in our word frequencies would lead to a higher weight on very probable responses, which could be viewed as beneficial if coherence of response is strongly desired. We also furthered this additional weighting by filtering out word possibilities with less than a 40% occurrence when compared to the most frequent word occurrence. Had we applied log statistics to our response selection, a wider set of words would be used for response generation. Another question was regarding our handling of unfamiliar responses. For words that were not identified in our corpus, a set of general responses that are not sensitive to the input context were used. While rules could have been created to handle unknown subjects or words given to the bot, we instead focused on future refinement of our bot. To do this, we store all inputs provided by the user so that the bot can learn from its userbase. These new user-created datasets are not implemented immediately so that they can be checked for authenticity and so that any undesirable utterances can be removed. Ultimately, despite our bot’s lack of coherence sometimes, we feel that we learned a lot about the strengths and weaknesses of probabilistic and more rudimentary approaches to language processing and specifically chatbots.

“An Intelligent Interface That Is Ready When You Are.” *Next IT*, www.nextit.com/case-studies/charter/. Accessed April 16, 2018.

Armstrong, Joseph. "Markov Chains - Explained" techeffigytutorials.blogspot.com, 8 Jan 2015 Web. 5 May 2018.

Chantarotwong, Bonnie. "The learning chatbot." *Berkely IMS-256 Final Project, http://courses.ischool.Berkely.edu (Fall 2006)* (2006).

“Cornell Movie--Dialogs Corpus.” Cornell Movie-Dialogs Corpus, [www.cs.cornell.edu/~cristian/Cornell\_Movie-Dialogs\_Corpus.html](http://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html).

Maltby, Henry. "Markov Chains" brilliant.org, 14 Jan 2014 Web. 5 May 2018.

Powel, Victor. "Markov Chains" setosa.io, 7 Nov 2014 Web. 4 May 2018.

Shah, Nilima. “Which Is Best for You: Rule-Based Bots or AI Bots? – Chatbots Magazine.” Chatbots Magazine, Chatbots Magazine, 2 Feb. 2017, chatbotsmagazine.com/which-is- best-for-you-rule-based-bots-or-ai-bots-298b9106c81d.

Soni, Devin. "Introduction to Markov Chains" towardsdatascience.com, 7 Mar 2017 Web. 3 May 2018.

Srivastava, Tavish. "Introduction to Markov chain : simplified!" analyticsvidhya.com, 17 Jul 2017 Web. 4 May 2018.

“TextBlob: Simplified Text Processing.” Contributing Guidelines - TextBlob 0.15.1 Documentation, textblob.readthedocs.io/en/dev/.