Sam Jijina

# Project 3 Reflection

Throughout project 3, I followed a strict and thoughtful algorithm in order to design an agent that can learn how to answer questions as accurately and intelligently as possible. In my algorithm, I first began by creating a framework for extracting information and finding links amongst various information. I then examined each user question and removed all grammar (stop-words) in efforts to parse the sentence as accurately as possible. Once the grammar was removed, a lookup as to whether the given label was already defined was performed. If it was defined, the bag-of-words was populated, and my stemming algorithm was run on each word in the bag compared to the words already defined. After the stemming stage, the base words and concepts were added to the dictionary. If the label was not defined, the same steps mentioned above would be carried out except stemming since there was no historical data to run comparisons against. After the bag-of-words was populated, analogical reasoning was used to attempt to find patterns amongst the training data sentences for the same label. These patterns could be word order, starting word, or certain word combinations. Once these patterns were found, they were stored in the dictionary along with the bag of words and the label. After these steps, the agent also added any other unique words which were linked to the label. For the testing stage, the agent used a floating point confidence counter. As the agent iterated over each question, first it removed the grammar from the question and then used the same model defined in the training stage to find patterns. The patterns found by the agent were then used to search the training data dictionary to find a possible match or close match. As matches were found, the confidence counter was incremented. If at any point the confidence reached above 90.0, the agent would keep a pointer to that specific label. If there were no other labels with high confidence, the label and confidence was returned by the agent. If there were multiple candidates for the label, then the bag-of-words for each label was retrieved and iteratively compared to find the closest match. This closest match was returned along with the confidence value. After each guess the agent made, the agent called a post-processing helper function which took in the training data and used the testing question to extract patterns. It then incorporated these patterns into the model. If the helper found that adding a pattern was over specializing, the helper did not add the pattern, but if the helper noticed that the pattern kept repeating, the agent added the pattern to the model, therefore making the agent "learn" by using incremental concept learning. Two block diagrams for my agent are shown below. Depicted in Figure 1 is the agent's process of making a model from data and storing that model. Figure 2 illustrates how the agent then uses the model to predict the correct corresponding label for the data provided by the user.

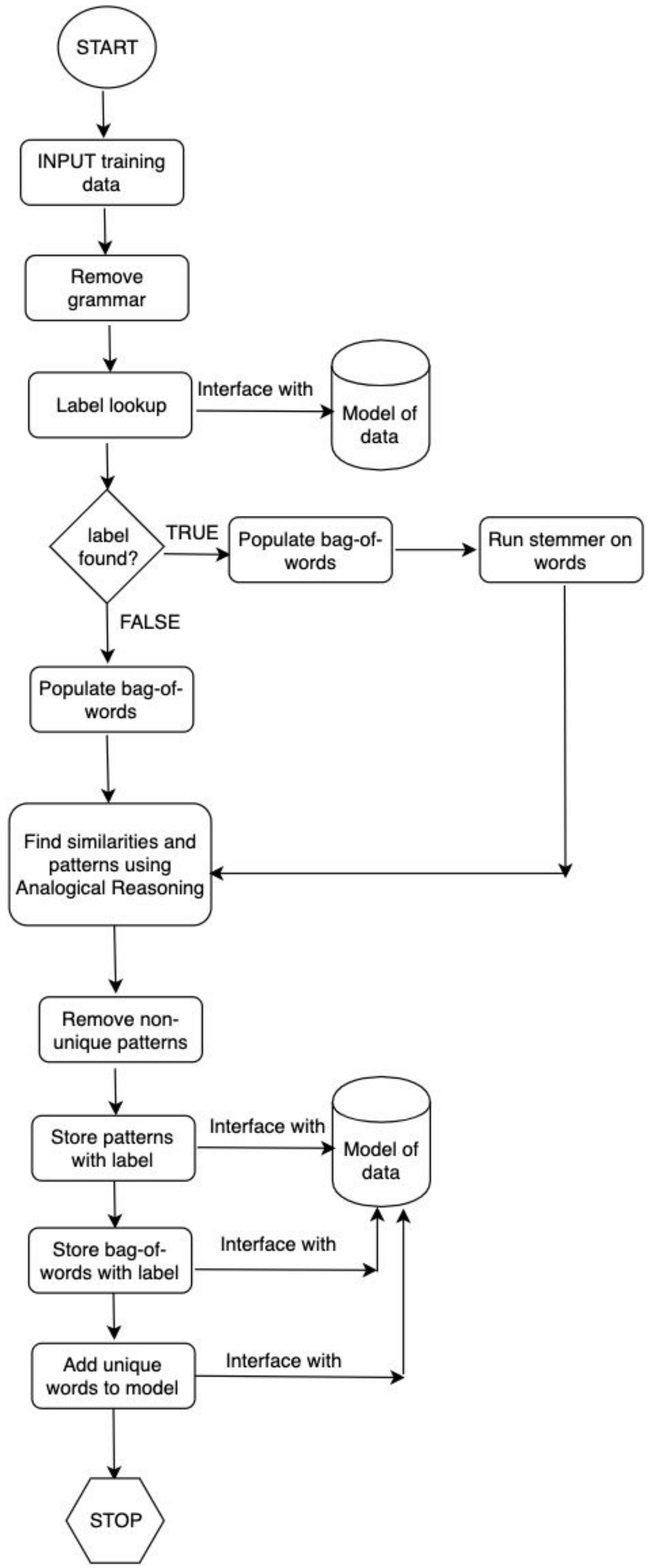


Figure 1: Agent Training

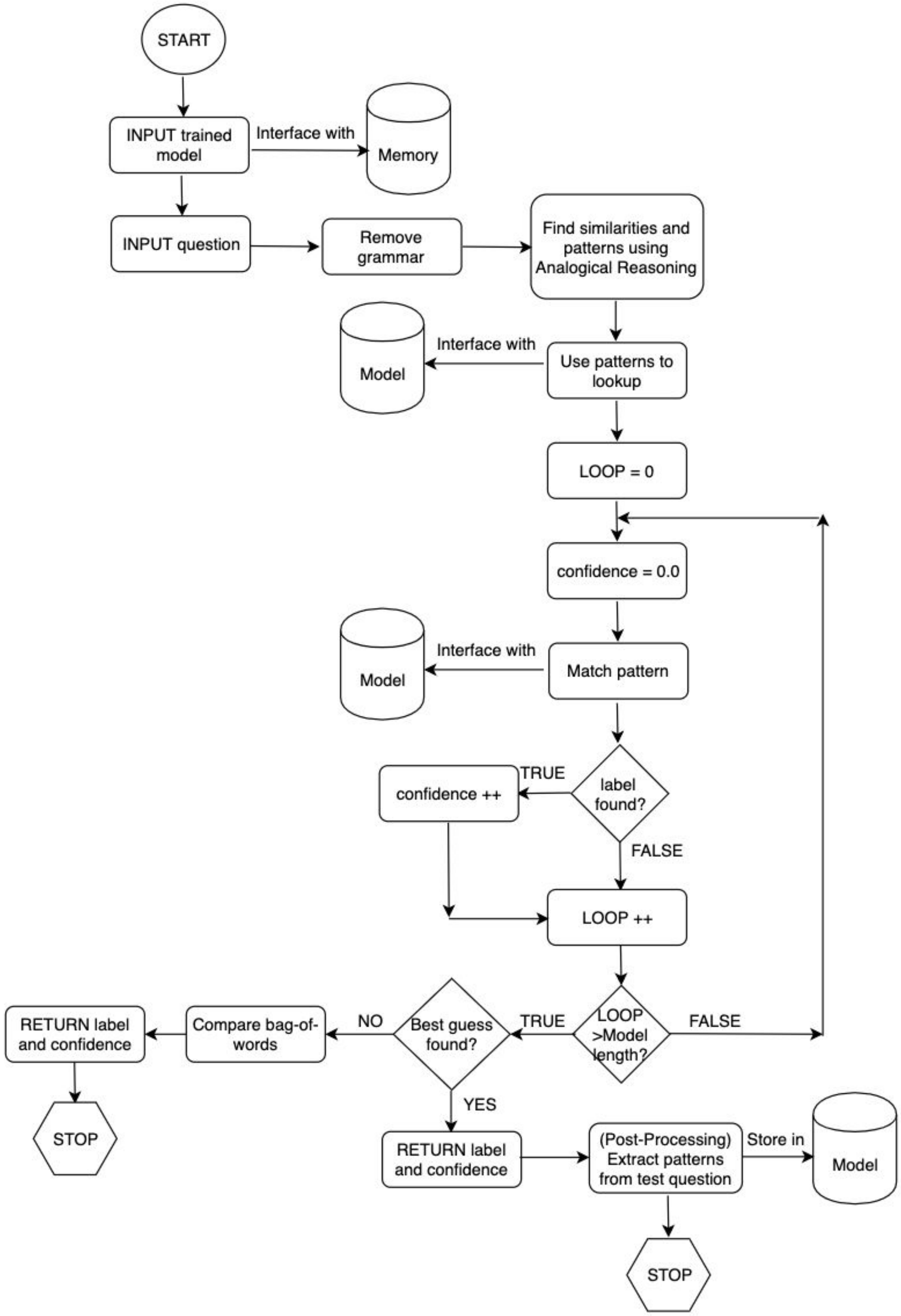


Figure 2: Agent Testing

For the design of project 3, I considered how human cognition enables us to identify patterns and created a model based on the data we come across everyday. Just as humans do, my agent learns when something doesn't match the present model and then modifies the model to improve accuracy. For example, the data "We use Piazza for this course" and the testing question "Where do I ask questions?" on the surface do not share any resemblance, but after the agent comes across the testing data, it can add new information to the already existing model. Therefore, the next time it is asked the same or a slightly similar question, the agent should be able to respond very efficiently and accurately. Another human-level cognitive ability shared by my agent is the ability to draw attention to outlier data, so if the agent notices a certain pattern is too different from existing patterns, it won't add it to the model as the agent is programmed to try its best not to overfit data. Additionally, the agent makes use of stemming, a concept which is also performed by humans as it involves identifying variants of a word based on an understanding of its root word. Certain unfamiliar words which humans come across are stemmed by our brains in an attempt to see if we can decipher the meaning by identifying the base of the word. However, this concept is more popularly recognized to be related to how machines are programmed to perform efficient information retrieval, and it proves to be a very efficient implementation for my agent's algorithm. Finally, analogical reasoning is an aspect of the agent that is arguably its most similar trait to human cognition. It helps both humans and my agent use analogies to make sense of various types of data that may have different meanings, but all involve similar ways of being interpreted and handled. Also, the agent uses analogical reasoning to check if the patterns given by incremental concept learning are already present and if so, the agent will not add those patterns, as that causes overfitting. In this way, just as learning and reasoning goes hand-in-hand with humans, it goes hand-in-hand with my agent as well. However, it is important to note that any task which requires a sense of intuition, emotion, or knowledge about entirely new domains is where the agent truly falls short of human cognition. This level of cognition is not currently possible by present day computing.

There were a few tradeoffs I made throughout the project in designing my agent. A tradeoff I faced was deciding to make an agent that was either highly specialised or generalised. A specialised agent would be more accurate on the data set given, but would fail on a completely new data set. A generalised agent is more accurate overall, but not nearly as accurate as a specialised agent on any specific data set. That being said, I chose to address this tradeoff by choosing to design the agent in a generalized manner. This design choice allows for the agent to answer a wider variety of questions more correctly than if I had designed the agent to be highly specialized, and this methodology proved to be much more successful in testing the agent for accuracy and efficiency.

The agent has several strengths in its design. Most notable is its ability to learn from training data which could be unrelated to a class, and then use testing data to make a confident guess, all the while learning from that guess itself. That being said, my agent is more robust than brittle as the bag-of-words approach along with incremental concept learning let it handle new and unfamiliar data sets while avoiding overfitting. The agent is able to successfully answer general and specific questions with high confidence levels and can adapt to new situations by implementing the new concepts it learns into the training model. Certain test questions which would utilize a completely different bag-of-words would still be answered correctly because their structure would form certain rules which the model would train on.

However, there are certain times when the agent fails to correctly identify the label from the given test question. This would occur if the training data question was too short or consisted of too many stop-words, and removing those stop words would leave behind less than four key words. The model found it very difficult to train on these kind of questions. However, since the agent uses bag-of-words as a fall-back method, the agent would still predict the correct label with high probability. Another issue that my agent faced was trying to differentiate between two very similar concepts. For example, “courseinfoteachingstaff” would be very similar to “coursecommunication”. This occurred because the training data for the two labels was very similar and when the agent gave these labels the same confidence level, the tie-breaker approach of using bag-of-words would also fail to choose one label over the other, sometimes resulting in incorrect performance.

Given an infinitely scalable vocabulary, the agent would certainly have limitations. It will perform accurately to an extent as the agent does not use vocabulary, but uses the hidden patterns behind the context and structure of English sentences. However, it is still not nearly as close to human cognition as humans use multiple processes at once to make a decision, whereas this agent uses learning and reasoning only in a sandbox-like environment.

Throughout project 3, I learned a lot about human cognition and the basic ideology behind how AITA works. Throughout the development process, I was challenged to delve deeper into how I think and how the human mind operates in order to continuously improve the intelligence of my agent upon it encountering failures in sentence classification. I am looking forward to using this new framework of artificial intelligence thought processes towards progress on other projects in various other AI courses.