Project Title: The Botinator Chatbot System
Project Final Report for CS 175, Winter 2015
List of Team Members:

Khoa Hoang, 28925467, khoah@uci.edu
Marvin Lowrance, 35567044, mlowrance@uci.edu

1. Introduction

For this project, we explored methods for building information and conversation-based chatbots, all done under the theme of Arnold Schwarzenegger. For the conversational component, which we call ArnoldBot, a third-order Markov chain was used to generate responses derived from training data consisting of dialogue from several Schwarzenegger films, speeches, and interviews. Additionally, regular expressions and semantic parse trees were pursued as a way of building the informational component, which we call ArnoldFan. ArnoldFan serves as a bot to answer questions about the life and movies of Schwarzenegger while ArnoldBot interacts with users by responding with an amalgamation of elements from Schwarzenegger's films, speeches, and interviews. The two components are tied together by essentially having ArnoldFan be a script that instantiates an instance of ArnoldBot, which we've implemented as a class. We produced two different models for ArnoldBot to present in user studies — Model A has safeguards against ineffectively constructed responses as well as the ability to formulate responses based on keywords in the user's message while Model B does not. We saw that this keyword selection function by itself was crucial in having the bot generate messages that bore some relevance to the user's messages. In the midst of building ArnoldFan, we discovered how many aspects there are to building a fully functional natural language chatbot with the ability to return accurate answers to a user query. We focused on the task of getting ArnoldFan to recognize certain question forms and then converted those questions to familiar templates prior to attempting to answer the question of the user. We came to realize that it was beneficial to allow for a relatively small grammar and see just how accurately we could form responses based off of that narrow grammar.

2. Problem Statement

This project addresses the problem of automating Q&A services as well as the challenge of building an artificial agent capable of simulating human conversation. Companies such as PayPal have deployed "virtual agents" for online customer service chats in the past to cut down on costs associated with managing human workers (Felix 2009). The conversational potential of chatbot systems has also been enthusiastically explored among computer scientists as well as hobbyists alike. The Turing Test is recognized as the gold standard for assessing a chatbot's capabilities in exhibiting conversational behavior indistinguishable from that of a human being. Having Arnold Schwarzenegger be the driving force behind the construction of these information and conversation-based components was a fun way for us to explore aspects of such systems as well as our own fandom of the bodybuilder turned actor-politician.

3. Related Work

Researchers in the field of Q&A chatbots have come up with designs based on matching user input with patterns that map to a range of "templated" responses, which would evolve over the course of a conversation (Quarteroni 2007). This ability to "evolve" responses relies on keeping track of the messages sent by the user. As a simple example, let's say that the user asks the following question: "What movies are they showing at the local AMC?". The Q&A chatbot would then reply with the appropriate response, only to have the user send the exact same question again. As a reply to this repeat message, the Q&A chatbot could send the same answer as before, but also clarify the user's questions (e.g., "Are these the kinds of movies you wanted to know about?"). Such a dialogue management model adopts what is known as the information state (IS) approach. A "lesser" model would adopt a finite state (FS) approach, which doesn't take previous messages into consideration. ArnoldFan uses the FS approach in its Q&A sessions as well as the technique of matching user messages with patterns that map to pre-coded responses.

ArnoldBot, which we've built with an FS approach in mind also, will implement a Markov chain model similar to Jason Hutchen's Megahal, a past succeeder of the Turing Test. Responses are formulated in Markov chains through a process of choosing a commonly occurring word that appears after, say, an initially selected state of four specific words from the user's message. Each ensuing state in the Markov chain contains this commonly occurring word, but will maintain a length of four words (the length of the state is what's referred to as the *order*). Megahal uses this four-word state arrangement, but we've chosen a three-word state for ArnoldBot for reasons discussed later on. Moreover, Megahal's responses are based off of keywords in the user's message, which mirrors what we've attempted for ArnoldBot as well. However, Megahal put two Markov chains to use in generating responses based on keywords, but the reason as to how and why this worked was lost on us, so we settled with using a single Markov chain.

4. Description of Technical Approach

Underlying the conversational component of our chatbot system (ArnoldBot) is a third-order Markov chain model. We've implemented the model with a defaultdict data structure from Python's collections module, and it serves to represent the "brain" of ArnoldBot. Each key in this defaultdict is a 3-tuple storing all trigrams, including the punctuation marks, found in the training text (hence, the *third*-order Markov chain) and the corresponding value is a list that holds all of the words that appear after each particular trigram. Note that since punctuation marks are also included in each key, it is imperative that the user also provide proper punctuation marks in their messages — namely, at the end of them —in order to receive more effective responses from ArnoldBot (though improper capitalization wouldn't be an issue). The list value corresponding to each key does not contain distinct words; rather, words appear in the list for however many number of times they appear after a given trigram in the training text. A key represents a state in the Markov chain, and the associated list holds all of the possible transitions from this state. The Markov chain is "unravelled" by randomly picking a word in the list for a given key, and naturally, words that appear more often will have a higher chance of being selected (i.e., it's as though these are higher-weighted transitions). The subsequent state then consists of the last two words from the previous state with the word that was just randomly selected. To illustrate this process, consider the following example:

. KEY (state): ("My", "name", "is"), VALUE (transitions): ["Arnold", "Arnold", "Bob,"]
—>
. KEY(state): ("name", "is", "Arnold"), VALUE (transitions): ["and", "Schwarzenegger", "I'll"]

ArnoldBot's responses are initially formulated by selecting a random trigram from the user's input (other measures are taken if the user's input is less than three words) to use as an initial state that "seeds" the Markov chain. For instance, let the user's response be "Hi, how are you?" with the selected initial state being ("how", "are", "you?"). The chain unravels in the fashion described above, concatenating the response with each word from each transition and stopping once a period, question mark, or exclamation point is encountered on a word that is transitioned upon. We observed that using one and two-order Markov chains tended to produce horrendously ungrammatical rubbish, but the third-order model was what gave us the most satisfactory outputs. We also tried out a fourth-order model, but discovered that the responses given by ArnoldBot were all too often exact lines from the training text. A fourth-order model could have been more viable if we had a much larger training text.

In order to give ourselves a chance at providing coherent responses for ArnoldFan, we had to restrict the grammar of the user significantly. To receive an accurate answer, the following criteria must be met:

(1) The user could only ask questions concerning Arnold or Arnold's movies.
(2) Questions could not be of the "yes" or "no" form.
(3) Questions about Arnold could only seek the date of certain events that took place in his life or the location of where such events took place.
(4) Questions about Arnold's movies can only seek to reference the date of when certain movies were released.

Given the relatively few questions ArnoldFan can answer, it is still an extremely complicated task to answer those questions being that there are so many ways a person could ask the same question. For example, the question "When is Arnold's birthday?" could be handled similarly to the question "When was Arnold born?" or "What month was Arnold born?". The first task in achieving such a feat would be to parse the user's question, extract the subject of the question, and determine what it is that the user is seeking to learn about the given subject. ArnoldFan was implemented primarily through semantic parse trees to parse a user's question and then return an accurate response. First, regular expressions are utilized to determine the nature of the question. The regular expression determines if the user is looking to discover "*where*" or "*when*" something occurred in Arnold's life. Upon discovering what type of question the user is asking, the sentence would then be tokenized. Each token is given a part of speech tag, and with that tag the subject of the question can be determined. Given the restricted vocabulary that we impose on the user in asking about Arnold or Arnold's movies, it was not too difficult to determine the subject matter of the user. All of the answers were stored in either Arnold objects or Movie objects. Based off of the question, a template is built from which the user's question could be easily answered. Given the question "Where was Arnold born?", the parse tree determines that the question is of the "where" form and the subject is Arnold. Next, the object template is passed with the question type being "where", question subject being "Arnold", and question info for the object being "born". The object builder then constructs an Arnold object with the parameter "where" and "born" filled in. Given the two parameters, the Arnold object then looks in the location attributes and determines whether or not the word "born" is something that it can determine, and if so it will respond with the proper sentence that answers the user's question.

5. Data Sets

Arnold movie scripts, interview transcripts, and speech transcripts were the data sets we used to train ArnoldBot. We first copy and pasted each script and transcript into their own files to keep track of what we were using and then compiled them all into a single master training text/corpus (found in the file "knowledge.txt"). A full listing of all the individual documents that make up the training text is included in the Supplementary.pdf file. The training text itself represents the knowledge of ArnoldBot, and it was important that we came up with a plentiful supply of text (i.e., knowledge) for two reasons: (1) So that the list associated with each key in the Markov chain (i.e., its "brain") would have many words, especially ones vital to proper sentence structure (like propositions), to transition on. (2) So that there could be a multitude of topics ArnoldBot could cover. Thus, we ended up copying and pasting all of the available Arnold movie scripts from the online movie script database www.springfieldspringfield.co.uk where roughly more than 80% of Arnold's movies listed on his imdb page were used. Since scripts were user-submitted, several were completely inaccurate and ridden with typographical errors. Such scripts were not put into the training text, as well as scripts belonging to films where Arnold merely played a cameo role. Transcripts for speeches that Arnold gave as governor and interviews from various entertainment, political, and bodybuilding magazines/websites were found through Google.

It was important that the training text was "cleaned" of elements that could potentially lead to extra-befuddling responses by ArnoldBot. First, we made sure that the training text had all lower case letters to avoid instances of inconsistent capitalization by ArnoldBot. Furthermore, the content of the movie scripts and interviews contain only the dialogue spoken — that is, there's no stage direction and there are no names of characters/individuals accompanying each line in the movies/interviews. Some of the movie scripts also had dashes and bullet points to indicate new lines of dialogue and, strangely, certain styles of symbols (e.g., quotation marks and apostrophes) that couldn't be decoded by Python. Trying to use the .read() method on File objects with these symbols led to a UnicodeDecodeError, which proved to be an extreme annoyance. The Python interpreter would fail to process the code if we didn't remove these symbols, meaning that we weren't able to move forward with processing the Markov chain until this issue was resolved. We also found the letter l in place of the letter i (lower-case l's can look a lot like capital i's) throughout a couple of movie scripts. These matters were dealt with and the unwanted marks were removed through the help of small Python scripts and our text editor's find and replace feature.

There were a variety of other issues in the training text that we fixed, which we found through our own chats with ArnoldBot. Typographical errors were a frequent issue, and it is more than likely that quite a bit still exist for branches of conversation that have yet to be explored. In addition, proper marking of punctuation was inconsistent in parts of the training text (e.g., several "how are you"s were missing question marks), an issue that if not rectified could cause less effective responses to be given by ArnoldBot. For example, since most of the training text is assumed to have mostly good punctuation, we recommend that users utilize proper punctuation marks in their messages to "hit" these well-punctuated states. However, when the user enters "how are you?" (note the proper punctuation), a "how are you" in the Markov chain that we overlooked in putting a question mark on wouldn't ever get hit, and there could perhaps be a particularly golden response that can be unravelled from there. Moreover, ellipses (…) were heavily put to use in the movie scripts, surely to convey the dramatic effect of the dialogue; however, these ellipses caused ArnoldBot to give rather awkward propositions that he'd more often than not fail to follow up on (e.g., arnoldbot: today we are going to play a game called…). The presence of the ellipsis, which is a string of three periods, causes the construction of a response to terminate on account of the period symbol being encountered. As such, many of these ellipses were removed, primarily the ones after conjunctions, prepositions, and verbs. Ellipses that were potentially able to pave way to sensible messages were kept (e.g., arnoldbot: sure glad we took care of business…). With regards to period symbols themselves, some were simply missing. The Markov chain stops unravelling based on where it sees periods, so if there isn't one where there should be, undesirable responses like "give me all of your money oh god don't hurt me shut up!" would arise.

The last notable modifications we made in ArnoldBot's training text involved actually writing new lines into it. This was necessitated by the appearance of simply awful responses to really simple questions that we would ask ArnoldBot while testing it out. For example, an ArnoldBot reply to "how are you" would sometimes be "that's odd reindeer are usually gentle." This made us cringe every time, so we changed it to something a bit more reasonable —

namely, “i feel unwell with you around”. More changes were made as we saw fit for any other appalling responses to simple questions.

The information needed to implement ArnoldFan is rather sparse and did not require many sources. The Arnold objects were given their information primarily through facts gathered off of www.imdb.com and www.wikipedia.com. We used relatively few data sets for ArnoldFan because the primary concern was figuring out how to accurately parse a question from a user and return an accurate response. The focus wasn’t on returning as much information on Arnold but rather on parsing user input and interacting with the user in a way that seemed as natural as possible.

6. Software

part (a)

All of our code is written in Python. ArnoldBot is represented as a class (appropriately called ArnoldBot) comprised of eight methods. Documentation is provided in the source code attached to this report, but a few things are worthy of note here. The __init__ method initializes an instance variable called _bad_first_words. This is a list of words that give particularly ineffective responses whenever they appear as first words in the response to be generated. We made these observations through our own chats with ArnoldBot. For instance, if “is” wasn’t in this list of words, ArnoldBot could respond to “what is your name?” with “is not a drill please proceed to the nearest exist”. This instance variable is used to make sure that such responses do not occur. We initially thought of having _bad_first_words be a list of English stop words, but discovered that it casted too wide of a net and limited the responses that ArnoldBot could give. _unravel_freq_tri picks a random frequently occurring trigram in the training text as an initial state (i.e., a seed) for the Markov chain and unravels a response from there. The unwanted variable is a set of trigrams containing states that, like _bad_first_words, give undesirable responses. The utility of this method will be discussed shortly.

The code behind the talk method is split up into two stages. **<u>The first phase</u>** selects a trigram from the user’s message to seed the Markov chain. If the user’s message only has one word, then that one word is essentially used as a keyword for selecting a state (a 3-tuple key) in the Markov chain. I.e., if that single word is “hi.”, then an element in the list of all states in the Markov chain with “hi.” inside of it will be randomly selected as the initial state in ArnoldBot’s response. If the message length is 2, then the longest of the two words is selected as a keyword and the same algorithm is carried out as for when the message length is one. In cases with a message length greater than 2, a random trigram is selected from the user’s message as mentioned before.

Before the second phase is carried out, a **<u>pre-processing phase</u>** for it is undergone. The first order of business here is to check if that state is even found in the Markov chain — i.e., if there’s a “state hit”. If it is, then the code checks if the selected trigram doesn’t appear “at the edge” of the user’s message. E.g., let the user’s message be “how are you doing?” and the selected trigram be “how are you” (note how there’s no question mark). This is an example of a trigram not appearing at the edge, in which case “how are you” would be appended to the response to be output by ArnoldBot to prevent an annoying “filling-in-the-blank” moment. E.g., *with this safeguard* in place, an ArnoldBot response could be “how are you doing i’m good.” *Without this safeguard*, a response could be “doing i’m good”. The former is much more preferable than the latter, which basically just finishes the sentence to “how are you”. The reason why we don’t append the words in an initial state to ArnoldBot’s response when the selected trigram is indeed on the edge is because *the end of the user’s sentence* is at the edge, and this would coincide with an end of a sentence inside the training text. After the end of a sentence inside the training text, a response to that sentence would be given, and so part of that response in the training text would be used for ArnoldBot's response. An ArnoldBot response to "how are you?" is often pretty good because of this arrangement.

The while loop that occurs after this preprocessing step represents the **second phase** where the Markov chain is unravelled from the the initial state. Inside the while loop is an if-else statement that checks for state hits. The check for a state hit is needed at the very beginning of the while loop because a state that isn't in the Markov chain could be encountered during the process of unravelling. In the event that a state isn't in the Markov chain, the _msg_by_keyword method is called. From here, a keyword is selected from the *entire* message of the user, and a state selection process similar to the first stage of the talk method is carried out. We've defined a keyword as the longest word in the user's message since the longest ones are potentially the most interesting. In addition, stop words were removed from the user's input as they could lead to an uninteresting selection for a state. If there doesn't exist a state in the Markov chain that contains the keyword, then a state from the approximately 50 most frequently occurring trigrams is selected (approximate because a couple of unwanted trigrams in this top 50 are discarded). The idea is that these frequently occurring trigrams could provide general responses serving as a sort of "catch-all" to any message that the user gives. For example, the top three trigrams are: 1. ('what', 'do', 'you'), 2. ('what', 'are', 'you'), and 3. ('i', 'want', 'to'). Any of these could easily serve as the basis for responses that are, say, evasive yet still relevant or that can veer the conversation away altogether from topics ArnoldBot has no knowledge of.

Once a full response has been generated, a final check on it, which can be viewed as a sort of **post-processing phase** of the talk method, is carried out. If the response generated has a "bad form" (e.g., the first word in it is a word in the _bad_first_words list), then the response is scrapped and another one is formulated based exclusively on a keyword in the user's whole message rather than a randomly selected trigram from the message. The reason why we don't simply do keyword selection right off the bat is because well-structured responses to certain messages — e.g., again using "how are you?" as an example — wouldn't be made. If we used keyword selection for these kinds of messages, then that single keyword selected from "how are you?" could generate a state that would have nothing to do with the semantics behind what all of those words mean together.

ArnoldFan is implemented primarily through object oriented programming. All of the classes we used to store information about Arnold's life and his movies are custom-made. The NLTK library is used to tokenize and tack on parts of speech tags to each token. The newly tagged tokens are then inspected within the proper module and it is determined as to which class will be referenced and on what attribute it is that we will be seeking an answer to give the user. Beyond tokenizing and giving each token a part of speech tag, all of the code is custom. There are essentially four python modules that each question must pass through in the process of returning an answer to the user. The user interacts with the interface, and given the form of the question, the interface will pass the question to the proper question type handler. The question form is determined using a regular expression. The form types include where, when, and what. Each type of question is handled differently. For instance, a "where" question is sent to a different module to be parsed than a "when" question. Within the given question form module the sentence is then parsed using the NLTK tokenizer and the NLTK part of speech tagger. The sentence will be inspected, and based on the parts of speech of that given token, it will be determined as to what the subject and interest are in each sentence. Once the proper information is derived from the parts of speech tags, a template will be formed and passed on to the sentence building module. The response builder module will then reference the appropriate class based off of the keywords in the provided template. If there is a match in the given class, the proper answer will be returned to the response builder. Given the answer and the form of the question, the response builder will build the proper response and return it to the interface so that it can be printed out on the screen for the user to see. If there is no match in the given class, an empty string will be returned to the response builder, which will then be returned to the question form module. Within the question form module a random response will be generated to inform the user that the answer could not be derived.

part (b)

The _build_brain method for ArnoldBot was based off of a code skeleton provided by Brian Chin on the website ActiveState while the outline for unraveling the Markov chain was found on a blog by Luke Dunn called Pythonism. The Python modules random, nltk, time, and collections were also put to good use in building ArnoldBot. The accompanying source code clearly presents how they are incorporated into the ArnoldBot class.

As stated before, the only outside software used in the implementation of ArnoldFan were two methods from the NLTK package, which were tokenize() and pos_tagger(). Such methods were instrumental in helping us break up the user's input so that ArnoldFan could decipher key terms from it.

7. Experiments and Evaluation

Evaluations and experiments for ArnoldBot came in the form of user tests. We were interested in seeing how well ArnoldBot Model A (MA), which used keyword selection and fill-in-the-blank safeguards, performed over ArnoldBot Model B (MB), which didn't have such features. Rather, Model B relied on unraveling a selected trigram from the user's message in case of a state hit or unraveling from a frequently occurring trigram from the training text in case of a state miss. Note that Model A is the one that has all of the features described in the previous "Software" section. Without telling the user which was which, we had them talk with Model A and Model B separately by starting off with a greeting (G) followed by a combination of six questions (Q1-Q6) and six comments (C1-C6). The questions and comments didn't necessarily have to be strictly sent in the order of six questions followed by six comments; users were allowed to mix up their six questions and comments in whatever way they saw fit based on the flow of the conversation. Users would then rate how much they liked each of the model's responses by giving a number between 1 and 3, with 1 being that they didn't like it, 2 being they sort of liked it, and 3 being they really liked it. The ideal score for each model was 39 (perfect 3's for all responses). We requested that each user let their rating be informed by how relevant each model's responses were to the message that the user sent and/or to the message that the model itself sent in the previous exchange. This latter case was for situations where the user would send a message and receive a totally random response in return, in which a 1 rating would probably be given. However, on the subsequent message sent by the user, the model could respond with a message that somehow relates to the totally random response that was given prior. The idea is that the ArnoldBot model is "too into his own thoughts" to reply properly to the user. Finally, users were asked to come into the conversation with the mindset that they were about to speak with an individual who "has a few screws loose". That is, the user should just play along with the ArnoldBot model since some of the zany things that it could have to say shouldn't be looked down upon too severely since it's just something that the bot can't help but do. At the end of each conversation, users were asked which model they prefer. The results, as well as chat logs, from seven user tests are captured in tables given in the accompanying Supplementary.pdf file.

Let us confirm that Model A outperforms Model B with a difference-of-means t-test. The following summary statistics are drawn from our results (subscripts denote the model): $\bar{x}_A = 28.29$, $\bar{x}_B = 20$, $s^2_A = 4.9$, $s^2_B = 5$. Assuming that the conditions for a t-test with $\alpha = 0.05$ and df = 6 hold on the population of all UCI computer science majors, we have the following null and alternative hypothesis: H_0: $\mu_A = \mu_B$ and H_1: $\mu_A > \mu_B$ where μ_A and μ_B are the true mean scores for models A and B, respectively. From the summary statistics we obtain a standard error value of 1.18. Computation of the t-score gives 8.47, with a corresponding p-value of 0.000074. With the p-value being much less than 0.05, we reject the null hypothesis of no difference in true mean scores. Our data therefore provides sufficient evidence that Model A performs better than Model B.

Upon allowing users to initially interact with ArnoldFan, we noticed that users were not asking questions that we originally intended to be asked of ArnoldFan. Often, users would venture off track and ask questions targeted more about ArnoldFan as opposed to actual questions regarding the life of Arnold Schwarzenegger. We realized how illogical it was to expect someone to ask ArnoldFan Arnold's date of birth five different ways on their own. We decided that it would be best if we not only gave users directions as far as the manner in which they can format their questions, but also found it necessary to inform the users of the topics that they could ask about in terms of Arnold's life and movies. After several unfortunate trials we decided that within each module based off of the question type, if ArnoldFan didn't have the proper answer or if it couldn't decipher from the question exactly what it is that the user is asking, it would generate a random response based off the form of the user's question. Assume for instance if the user were to ask the question "When was Arnold baptized?" Based off of the structure of the sentence, the semantic parse tree would send a template to the Arnold Date object with the term "baptized". Upon searching for the term "baptized", it would return an empty string simply for the fact that the Arnold Date object has not been supplied with any information in regards to Arnold being baptized. Upon the execution's return from the Arnold Date object with an empty string as the response, the "when was" module would then return a random sentence fitting for a "when was" question. A sample random response to a "when was" question would be "I don't know when that happened."

Evaluation of ArnoldFan was based primarily off of ArnoldFan's ability to render accurate results or simply just expected behavior. The primary goal was not to build a chatbot that users could enjoy interacting with, but rather a mechanism that is capable of parsing natural language and forming a response of the proper form. The majority of the tweaking came with recognizing the user's input and returning a statement that did not give an answer but illustrate the fact that ArnoldFan could at least understand the nature of the question. The primary issue with ArnoldFan is the restricted grammar and its inability to gather any type of meaning from a sentence with incorrect spelling. We learned that given the proper parsing, several sentences could be handled as if they were the same form. The questions "What is the birthplace of Arnold?" could be handled in the same manner as the question "Where was Arnold born?"

8. Discussion and Conclusion

For ArnoldBot, figuring out how to construct responses based on keywords from the user's input was a big step in getting ArnoldBot to speak in a manner somewhat consistent with the topics being discussed. There were numerous instances during user tests when Model A, unlike Model B, was able to win a 2 from the user simply because a mention of a keyword from his/her message was found in ArnoldBot's response. As expected for Model B, it *mostly* either triumphed with a 3 or flopped with a 1 in its ratings. There generally doesn't seem to be an abundant scattering of 2s across the scoreboard for Model B as for Model A. Moreover, Model A ended up scoring higher than Model B for each test, and all users, except for User2, preferred Model A over Model B. User2's justification for preferring Model B was that they felt this version of ArnoldBot appeared to have a more "consistent vibe" to it. This was perhaps one of the cases where the catch-all nature of the random frequent trigram state selection approach succeeded. Apparently, Model B had another noteworthy performance with User6 where it scored the highest out of any of its other tests (although it wasn't selected as the preferred model).

However, it appears that Model B *at its best* performed only about as good as Model A *at its worst*. Model A's best performance was given to a user who, compared with the other users, did a very good job of "playing along" with Model A as he tried diving into the specifics of what the bot was saying and worked off of what it was giving him. This makes us think about how much the effectiveness of Model A is contingent upon the user essentially letting the bot lead the conversation by introducing the topics to be discussed. This is perhaps because if the user continues to provide words that are in some way related to the keywords Model A had selected previously, then state (i.e., trigram) selections would correspond to particular sections of the training text pertaining to these sorts of terms.

Although it might be painfully obvious that Model A, with all of these additional features over Model B, must perform better than Model B itself, it was nevertheless interesting to see *how much* of an improvement we were able to get. As shown in the summary statistics, Model A almost did a whole tens digit better on average than Model B. In thinking about a model that could outperform Model A, we feel that we'd need to find a way to extract *meaning* from the user's message. Chapter 10 of the NLTK book provides ways of analyzing the meaning of sentences, but given the time constraints of the project and how complex some of the ideas in this chapter were, we decided to focus exclusively on the Markov chain for ArnoldBot. However, the idea in this chapter of using first-order logic as a way to extract meaning from sentences seems enticing and is something for us to consider on our own time. Furthermore, an informational state (IS) approach for ArnoldBot could be examined in the future. Such an approach would perhaps involve the use of a stack to store all of the messages sent during a conversation.

Attempting to build a somewhat functional natural language informational chatbot in the form of ArnoldFan proved to be a rather complicated task. There are several aspects one must take into consideration when building an informational chatbot. Taking in a sentence from a user and determining the proper form that your response should take is difficult simply due to the fact that there are so many ways you can phrase a statement and still maintain the same meaning. As you come to recognize forms of the same question, you learn that certain forms have a larger range of questions that can be asked, thus making it much more difficult to parse the user's sentence and give an acceptable response. Given the question "When was Arnold born", there is no problem in deciphering from the sentence that the sentence would require the program to search a Date object simply based on the first word of the sentence being "when". Parsing such a sentence is relatively straight forward and determining that the subject is "Arnold" and "born" is the information that the user seeks about Arnold is trivial. However, the question "what is the year Arnold was born" could ideally be answered in the same manner as the question "when was Arnold born". The issue is that it is not as easy to decipher what it is that the user is seeking to learn in that given form. The meaning of a question that begins with the term "what" is far more complicated to determine than one that begins with "when" or "where".

Given the complexity of deciphering certain sentence structures, it is a very daunting task to create an informational chatbot that is capable of interacting with a user in natural language. The standard approach to building informational bots seems to primarily involve deciphering the user's input and making a query to a database from which the bot would return an answer with the highest probability of being correct. There doesn't seem to be many informational chatbots out there that make queries and attempt to return an answer in such a natural language form that a user could easily feel the response was given by a human. Based on the experience of building a chatbot, it would be very interesting to see a major breakthrough as far as a highly accurate chatbot with an extensive vocabulary.

References

Chin, Bryan. “The Markov Chain Algorithm (Python Recipe)”. 2003. http://code.activestate.com/recipes/194364/

Dunn, Luke. “A Markov Text Bot in 71 Lines of Python”. 2010. https://pythonism.wordpress.com/2010/04/18/a-simple-chatbot-in-python/

Felix, Jerry. “AI and Python: Developing a Conversational Interface using Python”. 2009. http://www.slideshare.net/amyiris/ai-and-python-developing-a-conversational-interface-using-python

Hutchens, Jason. “How Megahal Works”. http://megahal.alioth.debian.org/How.html

Quarteroni, Silvia. “A Chatbot-based Interactive Question Answering System”. 2007. http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.122.1922

ARNOLDBOT USER TEST RESULTS

USER1:	G	Q1	Q2	Q3	Q4	Q5	Q6	C1	C2	C3	C4	C5	C6
MA	1	3	3	1	2	1	3	3	3	3	1	1	2
MB	3	3	1	2	1	1	1	1	2	1	1	1	2

Model A Score: 27/39
Model B Score: 20/39
USER1 preference: Model A

USER2:	G	Q1	Q2	Q3	Q4	Q5	Q6	C1	C2	C3	C4	C5	C6
MA	2	1	1	2	1	2	2	1	2	2	3	3	2
MB	2	2	1	2	1	1	2	3	1	2	1	1	1

Model A Score: 24/39 (*WORST SCORE FOR MODEL A*)
Model B Score: 20/39
USER2 preference: Model B

USER3:	G	Q1	Q2	Q3	Q4	Q5	Q6	C1	C2	C3	C4	C5	C6
MA	2	2	2	3	2	3	1	3	3	2	3	1	3
MB	1	1	1	3	2	2	2	1	1	3	2	1	2

Model A Score: 30/39 (*BEST SCORE FOR MODEL A*)
Model B Score: 22/39
USER3 preference: Model A

USER4:	G	Q1	Q2	Q3	Q4	Q5	Q6	C1	C2	C3	C4	C5	C6
MA	2	3	2	1	2	1	3	2	2	2	1	1	3
MB	1	1	1	3	1	2	2	3	1	1	2	1	3

Model A Score: 25/39
Model B Score: 22/39
USER4 preference: Model A

USER5:	G	Q1	Q2	Q3	Q4	Q5	Q6	C1	C2	C3	C4	C5	C6
MA	2	1	2	3	2	2	3	2	2	2	1	2	3
MB	1	1	2	1	1	2	1	1	1	1	3	2	1

Model A Score: 27/39
Model B Score: 18/39 (*WORST SCORE FOR MODEL B*)
USER5 preference: Model A

USER6:	**G**	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
MA	2	1	1	3	2	2	3	3	3	3	2	2	2
MB	1	3	2	3	1	3	2	1	1	1	3	2	2

Model A Score: 29/39
Model B Score: 25/39 (*BEST SCORE FOR MODEL B*)
USER6 preference: Model A

USER7:	**G**	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
MA	3	1	2	2	3	3	1	1	2	3	2	1	1
MB	3	1	1	1	2	1	1	3	1	1	1	3	1

Model A Score: 25/39
Model B Score: 20/39
USER7 preference: Model A

ARNOLDBOT CHAT LOGS

USER1
MODEL A:
You: hey dude! (G)
ArnoldBot: ain't the way to the surface. (G: 1)

You: how are you? (Q1)
ArnoldBot: hi come on 1 2, 3, 4. (Q1: 3)

You: what are you doing today? (Q2)
ArnoldBot: are you doing today fantastic. (Q2: 3)

You: yes i am fantastic. (C1)
ArnoldBot: it is fantastic to have my own line. (C1: 2)

You: you are smart. (C2)
ArnoldBot: know how smart you are. (C2: 3)

You: i am a genius. (C3)
ArnoldBot: i am a machine what is your name? (C3: 3)

You: my name is neeki. (C4)
ArnoldBot: you know what tobacco does to people. (C4: 1)

You: what does tobacco do to people? (Q3)
ArnoldBot: know what tobacco does to people. (Q3: 1)

You: no i don't know. (C5)
ArnoldBot: don't you need the willingness to work and do everything it takes to bring him in. (C5: 1)

You: who do i need to bring in? (C6)
ArnoldBot: i need to complete my research... (C6: 2)

You: what is 1 + 1? (Q4)
ArnoldBot: 5 to 1 on hercules to lose? (Q4: 2)

You: do you like coffee? (Q5)
ArnoldBot: like a vegetarian. really you can do it. (Q5: 2)

You: are you a vegetarian?
ArnoldBot: i don't want to know. (Q6: 3)

You: bye?
ArnoldBot: hasta la vista, baby! (Q6: 3)

MODEL B
you: how are you? (G)
arnoldbot: good you know julie how are you? (G: 3)

you: good thanks for asking. (C1)
arnoldbot: you have a sarah connor here? (C1: 1)

you: what are you doing today? (Q1)
arnoldbot: how are you what's going on here? (Q1: 3)

you: i am in class. (C2)
arnoldbot: one of the reasons you invited me here today was to honor my commitment to public service. (C2: 2)

you: what is your name? (Q2)
arnoldbot: favorite of your own movies? (Q2: 1)

you: what is your favorite color? (Q3)
arnoldbot: cigar great to meet you. (Q3: 2)

you: do you smoke? (Q4)
arnoldbot: do you want to be successful. (Q4: 2)

you: yes. (C3)
arnoldbot: put two more plates on it. (C3: 1)

you: what did you eat for lunch? (Q5)

arnoldbot: you have to have faith. (Q5: 1)

you: it is very warm today. (C4)
arnoldbot: i'm going to give you the greatest gift of all. (C4: 1)

you: what is it? (Q6)
arnoldbot: it i did i love you, john. (Q6: 1)

you: my phone is out of battery. (C5)
arnoldbot: the hourglass.. and i wanted to be a teacher we'll be fine. (C5: 1)

you: my cat is furry. (C6)
arnoldbot: do you want me to make him think... (C6: 2)

USER2
<u>MODEL A</u>
You: hello. (G)
ArnoldBot: hello. hey, vince, we haven't forgotten about you. (G: 2)

You: my name isn't vince. (C1)
ArnoldBot: isn't as good as you are here? (C1: 1)

You: what's your name? (Q1)
ArnoldBot: honey he doesn't want you to know that i'm still alive. (Q1: 1)

You: who's alive? (Q2)
ArnoldBot: we gave you lots of help. (Q2: 1)

You: yes, I need help. (C2)
ArnoldBot: gonna need it. his sixth cervical vertebra is crushed. (C2: 2)

You: are you a doctor? (Q3)
ArnoldBot: check his chin the doctor has fainted. (Q3: 2)

You: is somebody hurt? (Q4)
ArnoldBot: somebody over i want it back. (Q4: 1)

You: do you like pizza? (Q5)
ArnoldBot: pizza delivery guy is having a seizure. (Q5: 2)

You: the weather is hot today. (C3)
ArnoldBot: a hot stereo? stevie laughlin. (C3: 2)

You: you're very random. (C4)

ArnoldBot: you're already dead. everybody dies. (C4: 3)

You: you're not very nice. (C5)
ArnoldBot: you're still alive. and why is that? (C5: 3)

You: because i'm human. (C6)
ArnoldBot: human decisions are removed from strategic defense. (C6: 2)

You: what are feelings? (Q6)
ArnoldBot: our feelings any longer feelings? (Q6: 2)

You: bye
ArnoldBot: hasta la vista, baby!

__MODEL B__
you: hi. (G)
arnoldbot: i'll kill you you son of a bitch! (G: 2)

you: what are you talking about? (Q1)
arnoldbot: about i've been lookin' for you all night. (Q1: 2)

you: sorry, you have the wrong person. (C1)
arnoldbot: this is the wrong finger come here! (C1: 3)

you: no thanks, i'll be on my way now. (C2)
arnoldbot: not going to be close. (C2: 1)

you: lets talk about something else. (C3)
arnoldbot: that's really important today we're going to talk to you about mr hauser who are you? (C3: 2)

you: who's that, i don't know him. (C4)
arnoldbot: do you think that they're being arrogant, or misusing the sport of bodybuilding... (C4: 1)

you: you're going off topic now. (C5)
arnoldbot: do you know how it went. (C5: 1)

you: no, i don't know how it went. (C6)
arnoldbot: what he's like oh yes, i do. (C6: 1)

you: who's he? (Q2)
arnoldbot: looks rough don't start do you two need a moment alone? (Q2: 1)

you: where are you going with this? (Q3)
arnoldbot: my desk takin' back this furniture, man. (Q3: 2)

you: how about my chair? (Q4)
arnoldbot: i want to say the other two gentlemen won their height categories... (Q4: 1)

you: how tall are you? (Q5)
arnoldbot: give me the key and you won't tell me what's going on? (Q5: 1)

you: what is a key? (Q6)
arnoldbot: one of the richest men in the gym? (Q6: 2)

you: bye
arnoldbot: hasta la vista, baby!

USER3
<u>MODEL A</u>
You: hey, what's up? (G)
ArnoldBot: tickin' what's wrong, baby what's wrong? (G: 2)

You: what is wrong, baby? (Q1)
ArnoldBot: wrong with cereal? it's cliche. (Q1: 2)

You: are you attractive? (Q2)
ArnoldBot: attractive to work here i agree. (Q2: 2)

You: do you think i'm attractive, man? (Q3)
ArnoldBot: too attractive to work here. (Q3: 3)

You: the weather today seems scornful. (C1)
ArnoldBot: what are you talking about? (C1: 3)

You: i mean it's hot outside, man! (C2)
ArnoldBot: 20 miles outside sierra russo. (C2: 3)

You: do you go to vegas often? (Q4)
ArnoldBot: you go to the bathroom? (Q4: 2)

You: how do you feel about slot machines? (Q5)
ArnoldBot: how do you feel about women pursuing bodybuilding? (Q5: 3)

You: you got quite the sense of humor there, pal. (C3)
ArnoldBot: quite a speech. you heard that? (C3: 2)

You: what's one thing you would do to fix the world? (Q6)
ArnoldBot: what's taking so long barney ross? (Q6: 1)

You: your muscles are looking pretty manly, man. (C4)
ArnoldBot: my muscles stimulated and firm. (C4: 3)

You: there's lots of pretty girls out there today. (C5)
ArnoldBot: there's a quarter million dollars in that bag. (C5: 1)

You: terminator was a great movie man, good job in it. (C6)
ArnoldBot: the new terminator should be a perfect example, but thats what makes politicians fake. (C6: 3)

You: bye
ArnoldBot: hasta la vista, baby!

MODEL B
you: how are you, baby? (G)
arnoldbot: do you think the future of bodybuilding is? (G: 1)

you: pretty good, seems like people are getting healthier. (C1)
arnoldbot: do you want me to vacuum now? (C1: 1)

you: why would i want that? (Q1)
arnoldbot: you want to say where are you going? (Q1: 1)

you: where should i be going? (Q2)
arnoldbot: none of us are you're married. (Q2: 1)

you: no, i'm not, but i want to be one day. (C2)
arnoldbot: seen by my god mr.hobbes. (C2: 1)

you: i thought hobbes was evil. (C3)
arnoldbot: don't want to believe it. (C3: 3)

you: do you really believe what you want to believe? (Q3)
arnoldbot: you're telling me that i'm the first one to break that pattern. (Q3: 3)

you: i'm telling you there is no pattern. (C4)
arnoldbot: us you psychopathic bitch oh, sure. (C4: 2)

you: why am i a bitch? (Q4)
arnoldbot: i'm going to tell your wife, he - i mean, he couldn't believe it. (Q4: 2)

you: what he doesn't know won't hurt him. (C5)
arnoldbot: do you know what i'm sick of it! (C5: 1)

you: do you know your maker, khoa hoang? (Q5) arnoldbot: i want you to worry. (Q5: 2)

you: i am an android too. (C6)
arnoldbot: i want you all to consider yourselves picked up... (C6: 2)

you: where are you picking us up? (Q6)
arnoldbot: be off let me speak to conan. (Q6: 2)

you: bye
arnoldbot: hasta la vista, baby!

USER4
<u>MODEL A</u>
You: hi arnold! (G)
ArnoldBot: hi arnold!...? (G: 2)

You: are you not arnold? (Q1)
ArnoldBot: arnold braunschweiger schwarzenegger gesundheit may i help you? (Q1: 3)

You: do you love maria shriver? (Q2)
ArnoldBot: shriver good to see you. (Q2: 2)

You: she's not happy to see you. (C1)
ArnoldBot: young man lovely to see you all again. (C1: 2)

You: how old do you think i am? (Q3)
ArnoldBot: you think i look like? (Q3: 1)

You: how old are you? (Q4)
ArnoldBot: how old are you forget it. (Q4: 2)

You: i'll search it up on the internet later. (C2)
ArnoldBot: internet that use lines from your movies. (C2: 2)

You: i don't make movies. (C3)
ArnoldBot: movies they say make my day or i am your brother, julius. (C3: 2)

You: why did you want to be a politician? (Q5)
ArnoldBot: give me the give me the key! (Q5: 1)

You: which key do you want? (Q6)
ArnoldBot: melina she's busy but mary here, she's free. (Q6: 3)

You: i don't like your movies. (C4)

ArnoldBot: like letting me do something special... (C4: 1)

You: you suck as governor of california. (C5)
ArnoldBot: as governor of the great state of california, because i knew that every time i went to the funeral, and i said to myself, "okay, i'm gonna put this away. (C5: 1)

You: you are a republican. (C6)
ArnoldBot: republican and why i believe in this president. (C6: 3)

You: bye
ArnoldBot: hasta la vista, baby!

MODEL B
you: hi. (G)

arnoldbot: the last time we did it in a way that it is not fanciful to see california as the golden dream by the sea. (G: 1)

you: what do you like about the republican party? (Q1)
arnoldbot: your women blonde brunette redhead? (Q1: 1)

you: i only like blonde women. (C1)
arnoldbot: you have to think outside the box. (C1: 3)

you: no, blonde women really are the best. (C2)
arnoldbot: my cousin's sister's brother dug his way out there, when he was 5. (C2: 1)

you: he should've died in that hole. (C3)
arnoldbot: when i was and won my first trophy. (C3: 1)

you: you don't deserve any of your trophies. (C4)
arnoldbot: i got a car to take you places... (C4: 2)

you: i don't believe in bribery. (C5)
arnoldbot: i'm going to hang out at kelly's after work? (C1)

you: is kelly a lot more fun than maria? (Q2)
arnoldbot: do you think baxter was doing here? (Q2: 1)

you: baxter also hangs out with kelly a lot after work. (C6)
arnoldbot: i'm going to coordinate with the federal people. (C6: 3)

you: what do you do with kelly after work? (Q3)
arnoldbot: get out of the shower in less than an hour what a cutie. (Q3: 3)

you: maria takes a shower for more than an hour? (Q4) arnoldbot: what the hell quaid hey, quaid, you remember me? (Q4: 1)

you: can baxter join in the showers after work? (Q5)
arnoldbot: we're experiencing technical difficulties bullshit! (Q5: 2)

you: you would let technical difficulties get in the way? (Q6)
arnoldbot: car call n.y.p.d. get me an ambulance! (Q6: 2)

you: bye
arnoldbot: hasta la vista, baby!

USER5
MODEL A:
(MISSING **MODEL A** CONVERSATION. CONSOLE CRASHED)

MODEL B:
you: how are you today? (G)
arnoldbot: do you want to keep your friends. (G: 1)

you: do you have any friends? (Q1)
arnoldbot: a cure possible reinstatement get back into school. (Q1: 1)

you: are you arnold? (Q2)
arnoldbot: when i was a lawyer, a prosecutor. (Q2: 2)

you: how tall are you? (Q3)
arnoldbot: what do you say i'm training, arnold. (Q3: 1)

you: where were you born? (Q4)
arnoldbot: what do you want me to go through all of that just for a day of snowboarding? (Q4: 1)

you: can you say your abc's? (Q5)
arnoldbot: give me a break i got killed twice in two hours. (Q5: 2)

you: how many men have you killed? (Q6)
arnoldbot: one of the joys of being really young. (Q6: 1)

you: the weather is great. (C1)
arnoldbot: how do you tell that son of a bitch. (C1: 1)

you: that son of a bitch is me. (C2)
arnoldbot: cost me a career you brought in a suspect with half the bones in his body broken. (C2: 1)

you: yes i did. (C3)
arnoldbot: very strongly that the kids in the inner cities get disadvantaged with education, especially just recently when our governor and cruz bustamante has cut $120 million in textbooks which is supposed to be up here. (C3: 1)

you: hello governor. (C4)
arnoldbot: i enter this office beholding to no one, except you, my fellow citizens. (C4: 3)

you: sacramento is the capital of california. (C5)
arnoldbot: i don't know. right now iraq is a big surprise. (C5: 2)

you: it must be a big surprise. (C6)
arnoldbot: get out of here javed? (C6: 1)

you: bye
arnoldbot: hasta la vista, baby!

USER6
<u>MODEL A</u>
You: hello. (G)
ArnoldBot: hello. is dr abel in, please? (G: 2)

You: no, he is not here. (C1)
ArnoldBot: no, he is hopelessly insane. (C1: 3)

You: why is that? (Q1)
ArnoldBot: so i can just put the mail in! (Q1: 1)

You: what are you talking about? (Q2)
ArnoldBot: i'm only interested in her because she works at the school. (Q2: 1)

You: who is she? (Q3)
ArnoldBot: she's a polish peasant she fell into a trance about two weeks ago, and then she met mad dog ben richards… (Q3: 3)

You: did you go to poland? (Q4)
ArnoldBot: you go to the movies? (Q4: 2)

You: yes, do you go to the movies? (Q5)
ArnoldBot: movies like terminator and conan and true lies and predator and twins that went through the same problem. (Q5: 2)

You: who is the terminator? (Q6)

ArnoldBot: i'm a terminator listen to me very carefully. (Q6: 3)

You: you are stupid. (C2)
ArnoldBot: a stupid little plastic doll. (C2: 3)

You: a doll has more brains than you. (C3)
ArnoldBot: brains as well as the federal government, has a special responsibility. (C3: 3)

You: the weather is good today. (C4)
ArnoldBot: weather i'm late. but first, this. (C4: 2)

You: looks like you work out. (C5)
ArnoldBot: he looks at vaginas all day long. (C5: 2)

You: you look like a cool robot. (C6)
ArnoldBot: robot named turtle man turbo man. (C6: 2)

You: bye
ArnoldBot: hasta la vista, baby!

MODEL B
you: how are you? (G)
arnoldbot: good you know julie how are you? (G: 1)

you: when will the sun come up? (Q1)
arnoldbot: i don't know where they got it stashed. (Q1: 3)

you: i'm looking to rob a bank, are you in? (Q2)
arnoldbot: do you want me to do? (Q2: 2)

you: no, who are you? (Q3)
arnoldbot: we're the klane brothers that's burt, this is morris, and i'm bob. (Q3: 3)

you: can you help me pull off a job? (Q4)
arnoldbot: me one two three [cheering] look at this one now. (Q4: 1)

you: i'm talking about forty thousand, you in? (Q5)
arnoldbot: do you think it's time for you to renew your faith. (Q5: 2)

you: what are you on? (Q6)
arnoldbot: i don't want to break any hearts. (Q6: 2)

you: the weather is nice today. (C1)
arnoldbot: what are you doing let me get this... (C1: 1)

you: i won't let you go. (C2)

arnoldbot: in cold this gets old. (C2: 1)

you: it looks like you're alive. (C3)
arnoldbot: do you have a totally new chest now. (C3: 1)

you: you are a fine robot. (C4)
arnoldbot: oh.. shit yes [laughing] [vincent] that was spectacular oh, son of a bitch! (C4: 3)

you: this is cooking now. (C5)
arnoldbot: i don't want to do that. (C5: 2)

you: lets get down to business. (C6)
arnoldbot: this is a homemade explosive device! (C6: 2)

you: bye.
arnoldbot: hasta la vista, baby!

USER7
<u>MODEL A</u>
You: hello. (G)
ArnoldBot: hello. i'm your courtappointed virtual attorney. (G: 3)

You: how are you doing? (Q1)
ArnoldBot: we cannot allow this union to take place. (Q1: 1)

You: what kind of music do you like? (Q2)
ArnoldBot: been music to my ears come. (Q2: 2)

You: that's really cool. (C1)
ArnoldBot: that's a fine animal stay against the glass! (C1: 1)

You: do you drink beer? (Q3)
ArnoldBot: my protein drink with whiskey. (Q4: 3)

You: do you work out? (Q4)
ArnoldBot: work on your body and your business dealings, but what do you think it's even a little bit slower that you have to start with the midget. (Q4: 3)

You: that's dope man. (C2)
ArnoldBot: that's werner and that is all over for me now. (C2: 2)

You: do you like basketball? (Q5)
ArnoldBot: basketball at an allgirls school in rhode island, but... (Q5: 2)

You: i like basketball a lot. (C3)
ArnoldBot: i coached basketball at an allgirls school in rhode island, but... (C3: 3)

You: music is very interesting. (C4)
ArnoldBot: music has arrived. get mr morehart down here. (C4: 2)

You: lebron james is a great player. (C5)
ArnoldBot: i have a new chance. (C5: 1)

You: how do you like irvine? (Q6)
ArnoldBot: do you like better try it on. (Q6: 1)

You: i like irvine a lot. (C6)
ArnoldBot: i don't know what you're like. (C6: 1)

You: bye
ArnoldBot: hasta la vista, baby!

MODEL B
you: hello. (G)
arnoldbot: abel in please who's calling, please? (G: 3)

you: this is rohan. (C1)
arnoldbot: what the hell happened to you? (C1: 3)

you: i don't know, i had a tough day. (C2)
arnoldbot: this is a highsecurity facility. (C2: 1)

you: what are you doing right now? (Q1)
arnoldbot: don't know what i mean. (Q1: 1)

you: do you like soccer? (Q2)
arnoldbot: better try it on you don't mind waiting, i'll wrap it up. (Q2: 1)

you: what kind of clothes do you wear? (Q3)
arnoldbot: how do you know my name? (Q3: 1)

you: what is your name? (Q4)
arnoldbot: danny madigan where are you going? (Q4: 2)

you: will you be back? (Q5)
arnoldbot: i don't know what the problem is, but i am not deaf! (Q5: 1)

you: i like ice cream. (C3)
arnoldbot: you haven't been here all day, so you can't miss it. (C3: 1)

you: i like cigars. (C4)
arnoldbot: this is the vehicle's top speed. (C4: 1)

you: do you enjoy television? (Q6)
arnoldbot: how do you know about keeping your promises? (Q6: 1)

you: i try to do my best. (C5)
arnoldbot: you want to make an impression... (C5: 3)

you: i am happy. (C6)
arnoldbot: get out of the way! (C6: 1)

you: bye
arnoldbot: hasta la vista, baby!

ARNOLDBOT TRAINING DATA

Movie Scripts:
. 6th Day, The
. Batman and Robin
. Commando
. Conan the Barbarian
. Conan the Destroyer
. End of Days
. Eraser
. Escape Plan
. Expendables 2
. Hercules in New York
. Jingle all the Way
. Kindergarten Cop
. Last Action Hero
. Last Stand, The
. Predator
. Pumping Iron
. Raw Deal
. Red Heat
. Red Sonja
. Running Man, The
. Terminator, The
. Terminator 2: Judgment Day
. Terminator 3
. Total Recall
. True Lies

. Twins

Interviews:
. 60 Minutes Interview
. Reddit AMA
. Esquire Magazine Interview
. Fox News Interview
. Int Magazine Interview
. Oui Interview Interview
. Rush Limbaugh Interview
. Sean Hyson Interview
. Spiegal Interview

Speeches:
. Advice for Immigrants
. Capitol Alert Speech
. Conservative Party Address
. Emory University Commencement Speech
. Gubernatonal Debate
. Inauguration Speech
. Republican Convention Speech
. USC Commencement Speech
. Victory Speech

Miscellaneous:
. Random Arnold Quotes