

Chatbot Report

Purpose

The core idea of this chatbot is for the user to discuss their hobbies or interests with it. This is done by allowing the user to name Reddit communities (subreddits) that interest them, and the chatbot's responses are drawn from data based on those subreddits.

System Description

When the program starts, it will ask the user for a name. If the input matches the name of a previous user, the program will ask to confirm that the user is a returning user. If the user confirms they are returning with an affirmative statement, the program will continue, otherwise the program will ask the user to input a different name for themselves. To confirm whether a statement is affirmative or negative, the input is directly compared against hard-coded lists of affirmative and negative phrases. If instead the original name does not match any previous users, the program will welcome the user and continue.

Whether the user is returning or not, the program will give the user an opportunity to list some subreddits they are interested in. For each subreddit listed, the program scrapes the top posts from the subreddit in the past year and writes the data to file. This data consists of the original and lemmatized forms of all comments on all the scraped posts. If the subreddit data already exists on file, the scraping process will be skipped. This part of the chat will continue until the user reaches the limit of subreddits or they give the chatbot a negative statement.

Once the introduction process is over, the main loop of the program takes over. For each message the user inputs, the input message is parsed for various information about the user. Using the Spacy library, a dependency parse is performed on the input. Using this parse, it is determined whether the subject of the sentence is the personal pronoun "I." If that is the case, the predicate of the sentence is taken and compared against the synset for "like" to determine if the input resembles an "I like..." statement. If this is the case, and the predicate is not being negated, the objects of the sentence are taken and stored as the user's likes. If the objects happen to match the name of a subreddit, that subreddit is added to the user's preferences.

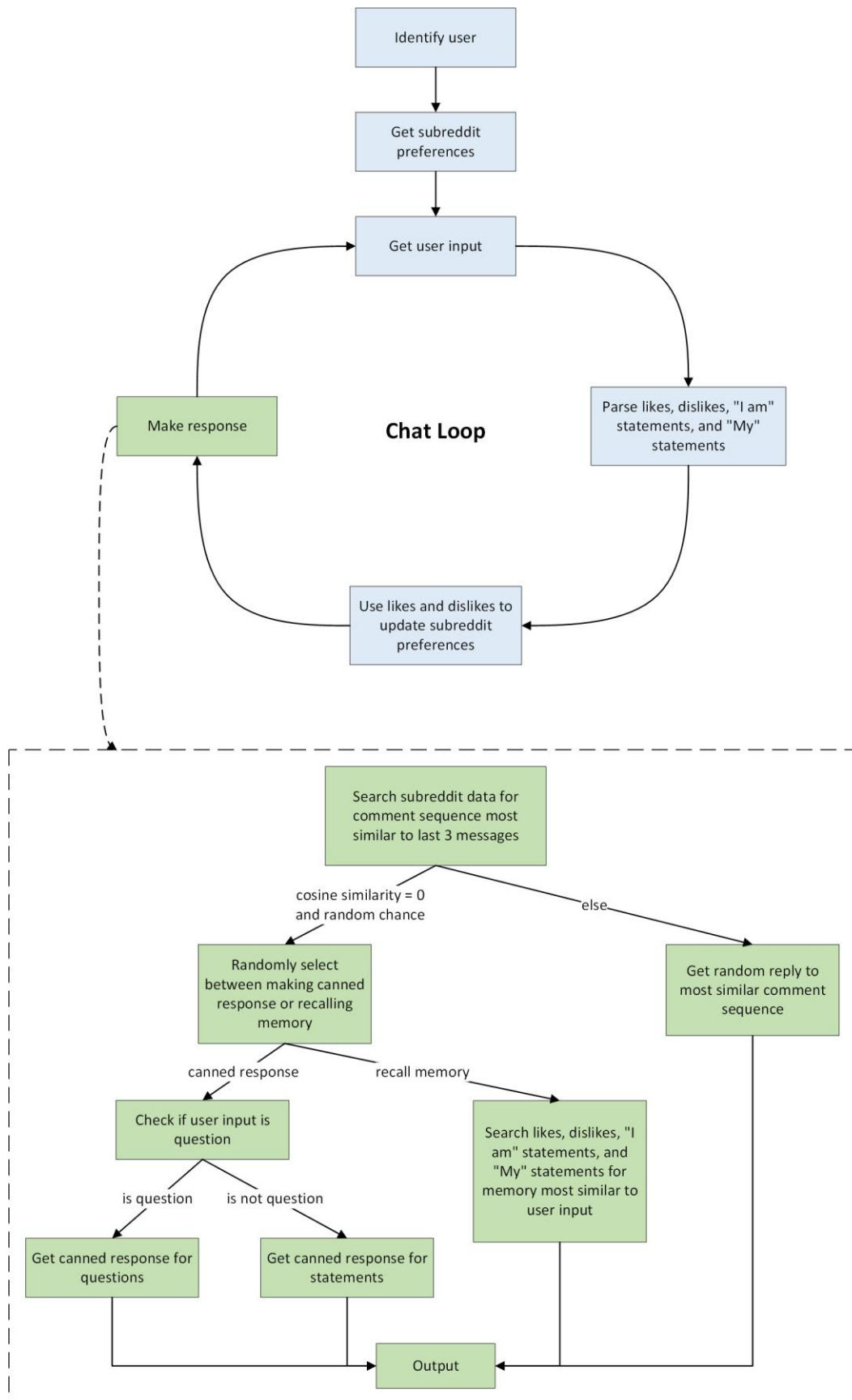
The predicate of the sentence is also compared against the synsets for "dislike" and "hate" to determine if the input resembles an "I dislike..." statement. The program also checks whether the predicate is a negation of a synonym of "like," as in the case of an "I don't like..." statement. In either case, the objects of the sentence are taken as the user's dislikes. If the objects match any subreddits in the user's preferences, these subreddits are also removed.

After parsing likes and dislikes with Spacy, the program parses “I am...” and “My... is...” statements from the input as well. For this purpose, simple regexes are used rather than a complex dependency parse. The phrases that complete each type of sentence are stored accordingly.

The program then attempts to search for a relevant response to the user’s input from the previously scraped data. To do this, the program first takes the last three messages in the chat history, including the user’s latest input. Call these M_0 , M_1 , and M_2 , where M_0 is the user’s most recent input, M_1 is the chatbot’s last message, and M_2 is the user’s second most recent message. A relevant response is found via a triple-nested loop. For every comment in the stored subreddit data, the program calculates the cosine similarity of the comment and M_2 . For each of the comment’s replies, the cosine similarity of the reply and M_1 is calculated. Finally, for each of the reply’s replies, the program calculates its cosine similarity to M_0 . The cosine similarities are summed and weighted to find the sequence of comments in the stored data that is most similar to the recent chat history. Once the most similar sequence is found, the chatbot picks a random direct reply to that comment sequence and outputs that as its response to the user. By considering the three most recent messages rather than just the user’s immediate input, the chatbot can retain the context of the conversation even if the user’s most recent message contains no domain-specific terms. Thus, even if the chatbot’s final output is not human-like, the response will usually come from the same source as its last response, retaining some sense of conversational flow.

If the largest summed cosine similarity of all comment sequences in the data is zero, then there is a random chance that the chatbot will respond with a canned response rather than with a comment in the data. In this case, the program will randomly select between returning a completely generic response or returning a statement about what the user has previously said about themselves. For the former, the program will determine whether the user’s input is a question by comparing it against a hard-coded list of question-indicative words. If it is a question, the chatbot will output a canned response appropriate to a question, and otherwise, the chatbot will output a more generic statement. If the program chose not to return a canned response, the program instead searches through its memories of the user for one that has the greatest cosine similarity to the user’s most recent input.

The main chat loop continues until the user inputs one of the specified exit words, at which point all user data is written to file and the program exits.



Sample Interactions

New user:

```
(Loading...)
Hello, what's your name?
$> andrew
Nice to meet you!
Name a subreddit you're interested in
$> formula1
Nice, anything else?
$> machinelearning
Okay, anything else?
$> technology
Nice, anything else?
$> programmerhumor
Got it, anything else?
$> iiiiiiiittttttttttt
Okay, anything else?
$> that's it
Got it! Let's move on...
(Processing...)
Alright I'm all ears now. What's on your mind?
$> my favorite language is C#
That's like, your opinion, man.
$> why, what's yours?
I'm not sureNot sure how to answer that
$> okay camelcase is annoying but it's not that bad
These things can take a lot of time to be fully apparent. Really hope Lewis
stays okay.
$> but what's your favorite language?
what do you think?
$> I mean you're written in python so...?
what are your thoughts on that?
$> quit
(Saving data...)
bye!
```

Returning user:

```
(Loading...)
Heya, what's your name?
$> Andrew
Sounds familiar. Have we spoken before?
$> think so
I thought I recognized you!
I remember your favorite subs now! Is there a new one you want to mention?
$> not really
Got it! Let's move on...
(Processing...)
Alright I'm all ears now. What's on your mind?
$> I don't really like formula1 anymore
I'll make a note that you don't like formula1 anymore.
```

```
$> so what was your favorite programming language again?
>!It wouldn't its a joke!<
$> what's a joke?
The thing that 113k people on reddit are now laughing at?
So seems to have worked.
```

Different user:

```
(Loading...)
Hello, what's your name?
$> Alice
Nice to meet you!
Name a subreddit you're interested in
$> Cycling
Noted, anything else?
$> Climbing
Nice, anything else?
$> Olympics
Got it, anything else?
$> travel
Cool, anything else?
$> That's it
Got it! Let's move on...
(Processing...)
Alright I'm all ears now. What's on your mind?
$> Have you traveled anywhere?
Why what? You don't hear many of those stories and sounds interesting if you're
willing to tell the tale. If not all good.
$> I've climbed a 9a before.
Ok it's not amusement but typically we can all learn something by hearing from
other people sharing their experiences. All good thank you.
$> What kind of cycle do you use?
what do you think?
$> What's your favorite Olympic sport?
A gri gri uses a camming mechanism to brake when the rope is pulled quickly. In
that picture Adam is holding the camming mechanism in place, so the gri gri as
held would not lock when the climber falls.
$> Have you scaled a cliff before?
Believe it or not there is more than one cliff in the UK. It's not even the
same stone/colour.
$> Cool bye
Interesting
$> exit
(Saving data...)
See you later!
```

Demonstrating canned responses:

```
(Loading...)
Hello, what's your name?
$> Bob
Nice to meet you!
```

```
Name a subreddit you're interested in
$> news
Okay, anything else?
$> sports
Noted, anything else?
$> cars
Got it, anything else?
$> dogs
Cool, anything else?
$> done
Got it!
Let's move on...
(Processing...)
Alright I'm all ears now. What's on your mind?
$> I like ice cream.
Ooh, now I know you like ice cream!
$> I don't like fish.
Huh, I didn't know you disliked fish.
$> Do you play basketball.
It's not impossible, and I'd love to see it. I'm just not gonna get my hopes up
too high on what is currently just speculation trying to make sense of a
confusing statement.
$> Do you play soccer?
Username fits?
$> I have a golden retriever.
I remember you said you liked ice cream
$> I like dogs and ice cream
Ooh, now I know you like dogs and cream!
$> I also like rainy days.
Ooh, now I know you like rainy days!
$> exit
(Saving data...)
cya
```

Evaluation

The chatbot of course is not very human-like. It is rare for the chatbot to make a response that naturally follows from the user's input. If the chatbot does make a good response to the input, it is usually just a happy coincidence. The chatbot is particularly bad at answering questions. This is a result of reddit comments not usually taking the form of one-on-one conversations but rather a series of related statements. This means the chatbot is poor at playing the role of a person with simulated thoughts and experiences. The chatbot is typically good at making responses that are relevant to the subject of discussion, but this is highly dependent on how focused the user's subreddit choices are. If the user decides to base the chatbot on many subreddits with different subject domains, the chatbot has a tough time appearing relevant due to how likely it is for a similar comment sequence to be found in an irrelevant subreddit.

The rules used for parsing user data are fairly effective and accurate. The program could be easily expanded to utilize the current user models in a more powerful way. It could be argued that the strategy of finding similar comments from a subreddit is not inherently flawed and that the chatbot could

perform better if it had access to more data. The current form of the program sets an arbitrary limit on how much of each subreddit is scraped for practical reasons. If more of each subreddit were scraped or if the chatbot could access other relevant webpages, the same strategy may result in more natural responses to the user input. However, this strategy is unlikely to ever overcome the issue of responding naturally to questions simply because most written language on the web is not in the form of turn-based conversations.

Data Appendix

The data used by the chatbot comes from the subreddits that are mentioned by the user. For each subreddit associated with a user, the chatbot visits the top 30 posts of all time plus the top 30 posts in the past year. For each post, the chatbot recursively scrapes the text of as many comments as possible. The data associated with each subreddit is written to separate files. When the program starts, it loads the data for a user's subreddits into a simple Python dictionary accessed by subreddit name.

For example, this would be how the data may appear for the interactions seen above (note that the program also stores if comments are replies to one another):

programmerhumor	That's like, your opinion, man.
	I'm not sureNot sure how to answer that
formula1	These things can take a lot of time to be fully apparent. Really hope Lewis stays okay.
news	It's not impossible, and I'd love to see it. I'm just not gonna get my hopes up too high on what is currently just speculation trying to make sense of a confusing statement.
climbing	A gri gri uses a camming mechanism to brake when the rope is pulled quickly. In that picture Adam is holding the camming mechanism in place, so the gri gri as held would not lock when the climber falls.
	Believe it or not there is more than one cliff in the UK. It's not even the same stone/colour.
travel	Why what? You don't hear many of those stories and sounds interesting if you're willing to tell the tale. If not all good.
	Ok it's not amusement but typically we can all learn something by hearing from other people sharing their experiences. All good thank you.

User Model Appendix:

For each user, the program stores their associated subreddits' names, likes, dislikes, "I am" statements, and "my" statements. When the user inputs a like or dislike statement, the program stores the object of the sentence as the like or dislike, respectively. Likes and dislikes for a user are each stored in their own sets. For "I am" statements, the program stores the string that completes the sentence in a set. For

“my” statements, the program creates a dictionary where the string between “my” and “is” is the key and the string that completes the sentence is the value.

The following are possible user models for the interactions seen above:

Andrew		
subreddits	machinelearning	
	technology	
	programmerhumor	
	iiiiiiitttttttttt	
likes		
dislikes	Formula1	
I am		
my	favorite language	C#

Alice		
subreddits	cycling	
	climbing	
	olympics	
	travel	
likes		
dislikes		
I am		
my		

Bob		
subredits	News	
	sports	
	cars	
	dogs	
likes	ice cream	
	dogs and ice cream	
	rainy days	
dislikes		
I am		
my		