Report on ChatBot

2021 Summer KNU A.I. & NLP

2016115153 Jaehong Ju

1.   introduction

This project is made by Java, accepts inputs and prints outputs via standard IO (System.in/out).

 - "ChatBot.java" contains main function.

 - "Listener" class is for analyzing inputs from user (or could call "customer").

 - "Speaker" class is for print proper answer according to results of analysis of Listener.

 - Program also includes "dict.txt" that saves several words.

 - Github : https://github.com/wnwoghd22/ChatBot

For executing, on command prompt type "javac *.java & java ChatBot" (JDK is needed)

2.   on ChatBot program

This ChatBot is a program that accepts natural sentences from user and finds proper answer, especially in situation occurred in marketplace. Because it only contains simplest logics and small dictionary, sentences that can be answered are not general. In detail, Listener only checks whether sentence includes a verb or not but does not check more than two verbs. It is same for noun and adjective. Dictionary that used in program contains only 7 verbs (3 are "am, are, is"), so user can only use 5 verbal lemmas.

For example, when user says, "I want to buy fruit." Then, listener finds two verbs, "want" and "buy", but only second one is accepted and answers, "What kind of fruit do you buy?". It is unnatural.

Listener has several functions and uses pattern matcher in that functions. While tokenizing, Listener uses Scanner to separate each word (but only by white space, tab, or new line etc.) and checks which POS the word is. (Actually, it is not a complete tokenizing – It cannot catch apostrophes like I'm or you're and also separate proper nouns like New York to several tokens, etc.) Checking if sentence includes "I" is relatively simple, it is only written in uppercase. but checking "You" is slightly complex, because if "you" is placed at start of sentence, it becomes "You", but in the middle of sentence, "you". So, person checker used regular expression matcher (method "matches(String RegEx)" which is predefined in java.String class). Down below is pseudo code.

if(token.matches("^[Yy]ou$")) then { set Second-Person(); } // if sentence includes You or you

And because of the incomplete tokenizing process, word checking also needs pattern matcher. A word can be separated from any other components (it's the best) but can by followed by punctuations. For example, word at the end of sentence is separated by Scanner, but it contains period(.) or question mark(?). Then String.equals() method may return wrong value. How about String.contains()? It is more likely returns wrong answer. If word checker finds word "is" from radish (or even "table" from "vegetable"), it returns true! But these are completely different. Instead, word checker used pattern matcher, ignoring last one character.

if(token.matches(word + ".?")) then { Find Keyword(); } // + ".?" : one character may followed or not.

Let us see the result.

First, ChatBot ask user to give help, as a default value. Default asking question is selected randomly from three phrases ("What can I help you?", "How may I help you?", "Is there anything you find?").

Then, if user says "categorical" keyword (in this picture, "fruit"), bot asks again, makes user answer more precisely. Or, if user says "item" keyword (apple, in this case), ChatBot asks user to say attribute. Finally, when user says specific item and attribute, then ChatBot answers as closing conversation. (Randomly choosing a phrase from: "Good choice.", "<random price> dollars, please.", "We don't have such item.", "Already sold out. Sorry.")

If Listener did not find any keyword, Speaker answers randomly from several pre-coded sentences like third line of picture above. ("I wouldn't understand what you say.", "Pardon me?", "What did you say?") And for some phrases if user says "I", then ChatBot answers using "You", and vice versa. But Listener just checks only "you" or "I', not both. So as like 6 and 7 lines above ("I want you to sell fruit." – "Which fruit do you sell?"), ChatBot answers unnaturally. Also, ChatBot only accepts at most one verb, so full verbal phrase ("... want to buy...") is not repeated, but just last one.

Listener checks if sentence contains string "leaving" at first, as exit protocol.

3. Discussion

Obviously, this program has lots of problems.

- Tokenizer only separates tokens by white space. It makes hard to catch abbreviated forms ("I'm", "You're", ...) and may separate proper noun.

- Word checker (It performs the role of Tagger, but not complete at all) checks words with only small .txt file. It only checks words with "BASE form", all inflected forms (even plural noun) are ignored except "be" verb.

- ChatBot only remember one keyword and one verb. It cannot repeat verbal or noun phrase, but only one word instead. And ChatBot only "echoes" verb except be (am - are). It is acceptable for English, but probably is not for other languages. (For example, "vendo – vendes" in Spanish, or "komme – kommst" in Germen)

- Structure of dict.txt is like this: <word> <POS> <isCategorical>. Because this assignment is quite simple, dictionary only contains POS of word and one information: if this word is categorical or not. Dictionary is needed to be improved.