Pavan Kumar Kothagorla
UIN: - 675007844
SandeepKumar AnilKumar
UIN: - 650005255

CS 421 – Natural language processing – spring 2016

Team Project: Building Mini-Watson (Part 1) Report.

Description: -

So we have pkotha6_sanilk2.java, which has the main logic of the working of the first part of this project. Inside this program, we are first taking a user input of the text file that should contain the questions. Using that argument in the main() function we start parsing the questions one at a time. We are using InputStreamReader and BufferedReader to access the contents of the file. We build a list of those questions inside the list "questionsList". Using this as an argument to the buildParseTrees() function.

We start iterating on each question, and then call parse() on the String "question" and parse it using the Stanford CoreNLP parse function. Inside the parse function we set the properties for annotators as "tokenize, ssplit, pos, lemma, parse." Using the CoreNLP pipleline and annotation on the question text, we build a parse tree of the question. This function returns a List of Trees.

Then we call ner() function with the input as the “question". Setting the property in ner() function to "tokenize, ssplit, pos, lemma, ner". Using the CoreNLP pipleline and annotation on the question text, we build a ner of the question. This function returns a List of String.

After the control comes back to the buildParseTrees() function, iterating on the List of trees returned by the parse() function we build a String of parse tree called "tree". Similarly iterating on the List of String returned by the ner() function, we build a string of named entities called "namedEntities". Using the "tree, namedEntities and currentQuestion" as input, we call the function getDomain().

In the getDomain() function we have used a property file called "Keywords.properties", this file has key value pairs of keywords related to "music and movie" domain. We first split the question by spaces. And using each individual String in the question we check if they belong in the property file. When a match is found, the loop exits. The logic to determine the domain is as follows: -

a) If our "namedEntities" has "LOCATION", then we check if it has "PERSON" too. Because suppose the question can be "Was Beyonce born in USA". A named entity recognition will return "PERSON" for "Beyonce" and "LOCATION" for "USA". So this belongs to a MUSIC domain. Since we aren't using any Database right now to rightfully disambiguate between Movie and Music domain, we will be returning the domain as "MOVIE or MUSIC." Since the PERSON can also be a Movie star.

b) If the "namedEntities" does not have any "LOCATION" and has "PERSON", this clearly means that it belongs to the domain "MUSIC or MOVIE".

c) Similarly if the "namedEntities" has only "LOCATION" this means it belongs to the "GEOGRAPHY" domain.

d) If there is no useful "namedEntites" returned, then we check the keywords in the property file to see if any of the question keywords maps to the MUSIC or MOVIE domain.

e) If not, then we check if the pattern matches the "patternGeo" i.e. ".*\\bJJS\\b.*\\bNN\\b.*", this basically means that for any number of words, if there is an "adjective" at the start of the boundary of the word, followed by a "Noun", then it should belong to the "GEOGRAPHY" domain. e.g.: - Which is the highest mountain? This is just a workaround, because, if there was some adjective on a person or any other noun, and if that was not matched using the above rules from a) to d), then it should belong to the "GEOGRAPHY" domain. Again there might be some corner cases, we have kept out logic working for the given questions and the others alike.

f) If none of the above matches, we are looking for the POS "VBD" (Verb, past tense) in the tree. If there is any VBD then it should most likely belong to the "MUSIC or MOVIE" domain. As said above, if this domain was not recognized before as belonging to the "GEOGRAPHY" domain because of the rule c) and e) it is very unlikely to be part of the rule f).

Lastly, we return the domain from this function and print the parse trees and domain as well as the question in the format as asked.

Why we used Stanford CoreNLP Parser: - I and my teammate have worked for years on Java. And there are some features which we have worked on using the Stanford CoreNLP parser before, for e.g. in the Homework 2 & 3. So we found it easy to use Stanford CoreNLP Parser. Also there are some features which we wanted to use in this part 1, like “Named Entity Recognition.” And since there is a simple option to use this feature, we wanted to use Stanford Parser. And also we don’t have to give the grammar in Stanford CoreNLP. And we are also thinking of using this project as a Web Service in future, and CoreNLP provides support to run as a simple web service.

All these reasons compelled us to use Stanford’s CoreNLP Parser.

Pavan Kumar Kothagorla
UIN: - 675007844
SandeepKumar AnilKumar
UIN: - 650005255

Parse Trees: -

1c) Is the Pacific deeper than the Atlantic?

Sol: - (ROOT (SQ (VBZ Is) (NP (DT the) (NNP Pacific)) (NP (NP (JJR deeper)) (PP (IN than) (NP (DT the) (NNP Atlantic)))) (. ?)))

1e) Did Swank win the oscar in 2000?

Sol: - (ROOT (SQ (VBD Did) (NP (NNP Swank)) (VP (VB win) (NP (NP (DT the) (NN oscar)) (PP (IN in) (NP (CD 2000))))) (. ?)))

1f) Is the Shining by Kubrik?

Sol: - (ROOT (SQ (VBZ Is) (NP (DT the)) (NP (NP (VBG Shining)) (PP (IN by) (NP (NNP Kubrik)))) (. ?)))

1j) Does the album Thriller include the track BeatIt?

Sol: - (ROOT (S (VP (VBZ Does) (SBAR (S (NP (DT the) (NN album) (NN Thriller)) (VP (VBP include) (NP (DT the) (NN track) (NN BeatIt)))))) (. ?)))

2a) Who directed Hugo?

Sol: - (ROOT (SBARQ (WHNP (WP Who)) (SQ (VP (VBD directed) (NP (NNP Hugo)))) (. ?)))

2b) Which is the scary movie by Kubrik with Nicholson?

Sol: - (ROOT (SBARQ (WHNP (WDT Which)) (SQ (VBZ is) (NP (NP (DT the) (JJ scary) (NN movie)) (PP (IN by) (NP (NP (NN Kubrik)) (PP (IN with) (NP (NN Nicholson))))))) (. ?)))

2f) In which continent does Canada lie?

Sol: - (ROOT (SBARQ (WHPP (IN In) (WHNP (WDT which) (NN continent))) (SQ (VBZ does) (NP (NNP Canada)) (VP (VB lie))) (. ?)))

2h) With which countries does France have a border?

Sol: - (ROOT (S (PP (IN With) (SBAR (WHNP (WDT which)) (S (NP (NNS countries)) (VP (VBZ does))))) (NP (NNP France)) (VP (VB have) (NP (DT a) (NN border))) (. ?)))

2m) Where was Gaga born?

Sol: - (ROOT (SBARQ (WHADVP (WRB Where)) (SQ (VBD was) (NP (NNP Gaga)) (VP (VBN born))) (. ?)))

Pavan Kumar Kothagorla
UIN: - 675007844
SandeepKumar AnilKumar
UIN: - 650005255

2n) In which album does Aura appear?

Sol: - (ROOT (SBARQ (WHPP (IN In) (WHNP (WDT which) (NN album))) (SQ (VBZ does) (NP (NNP Aura)) (VP (VB appear))) (. ?)))

2o) Which album by Swift was released in 2014?

Sol: - (ROOT (SBARQ (WHNP (WHNP (WDT Which) (NN album)) (PP (IN by) (NP (NNP Swift)))) (SQ (VBD was) (VP (VBN released) (PP (IN in) (NP (CD 2014))))) (. ?)))

Category code: -

```
public static String getDomian(String ner, String currentQuestion,
String t) throws IOException
      {
            if(ner == null || currentQuestion == null || t == null ||
ner.length() == 0 || currentQuestion.length() == 0 || t.length() ==
0) {
                  return "Question is empty";
            }
            InputStream ir =
ParsingQuestions.class.getResourceAsStream("/com/resources/Keywords.
properties");
            Properties prop = new Properties();
            prop.load(ir);
            ir.close();

            String patternGeo = ".*\\bJJS\\b.*\\bNN\\b.*";
            String questionParts[] = currentQuestion.split("\\s");
            int partsLength = questionParts.length;

            Set keywordsSets = prop.keySet();
            String domain = "";

            for(int i = 0; i < partsLength; i++) {
                        if(!
(keywordsSets.contains(questionParts[i]))) {
                              continue;
                        }
                        else {
                              domain =
prop.get(questionParts[i]).toString();
                              break;
                        }
                  }

            if(ner.contains("LOCATION"))
            {
                  if(ner.contains("PERSON"))
                  {
```

Pavan Kumar Kothagorla
UIN: - 675007844
SandeepKumar AnilKumar
UIN: - 650005255

```
                        return "MOVIE or MUSIC";
                  }
                  else
                  {
                        return "GEOGRAPHY";
                  }
            }
            else if(ner.contains("PERSON"))
            {
                  return "MOVIE or MUSIC";
            }
            else if(domain != null && domain.length() != 0)
            {
                  return domain;
            }
            else if(t.toUpperCase().matches(patternGeo)) {
                  return "GEOGRAPHY";
            }
            else if(t.contains("VBD"))
            {
                  return "MOVIE or MUSIC";
            }
            else {
                  return "Could not find";
            }
}
```