Convert Human-Bot JSON Data

January 14, 2019

0.1 Conversational Intelligence (ConvAI) Challenge Human-Bot Data Conversion

Datasource: The human-bot chats stored as json files are downloaded from here: http://convai.io/data/ The datasets are generated as part of Conversational Intelligence (ConvAI) Challenge done under the scope of NIPS 2017 Competitions track.

Output: The **json file contains numerous one-to-one dialogues. We need to extract the conversation from each dialogue and mark both participants with different symbols per line.** The conversation data after extraction is encoded and fed to the model.

```
In [19]: #import packages
         import pandas as pd
         import json
         import os
         import csv
```

```
In [20]: #defines permissible characters for the chatbot

         alphas = 'abcdefghijklmnopqrstuvwxyz1234567890 .,?'
         alphas = alphas + alphas.upper()

         def permissible_chars(word):

             for char in word:
                 if char in alphas:
                     return True

             return False
```

```
In [21]: # Defining parameters.

         # CHANGE the File name to process different files. We have only 3
         file_name = "data_intermediate"

         infile = open("json/"+file_name+".json", "r")
         outfile = open("data/"+file_name+".yml", "w")

         # Get the JSON data.
         json_parsed = json.loads(infile.read())
```

```
In [22]: # Process the parsed json data to get 'text' tag from dialogue
         # This represents the conversation between 2 parties involved in dialogue
         chat= ""
         for i in range(0, len(json_parsed)):

                 dialog = json_parsed[i].get('dialog')
                 for j in range(0, len(dialog)):

                         text = dialog[j].get('text')

                         # From the data it is known that "End" is used as stop word
                         # for each dialogue or conversation between two people.
                         # Stop the iteration if the word "End" is found.
                         if (text.find('end') != -1 or text.find('End') != -1):
                                 break
                         if (text == 'start'):
                                 continue

                         # remove all tokens that are not alphabetic
                         words = [w for w in text if permissible_chars(w)]
                         conversation = ''.join(word[0] for word in words)

                 # mark each participant with some symbol.
                         # alternate b/w question and answer
                         if j % 2 == 0:
                                 chat += "- - "
                         else:
                                 chat += "  - "

                         chat +=  conversation + "\n"
```

```
In [23]: # Write output files as yml files.
         # print(chat)
         outfile.write(chat)
         outfile.close()
```