COMP 262
Assignment #2 – Chatbots

Due Date: Week #9 (**March-18-2022 at 11:59 PM**)

Purpose:
The purpose of this Lab assignment is to:

1. To gain hands-on experience in building conversational agents (chatbots) using ready-made APIs.
2. To carry out pre-processing and build chatbot components using deep learning.
3. To train and test a simple chatbot to serve a business process.

General Instructions:
Be sure to read the following general instructions carefully:

1. Exercise #1 of this assignment is group work.
2. Exercise #2 of this assignment must be completed individually by all the students.
3. Only provide the requested screenshots and make sure to have a complete screenshot, partial screenshots will not earn any marks.
4. You will have to add all the analysis and screenshots in the Analysis report.
5. You will have to provide a **demonstration video for your solution** and upload the video together with the solution on **eCentennial** through the assignment link. See the **video recording instructions** at the end of this document.
6. In your 5-minute demonstration video (Only required for exercise#2) you should explain your solution clearly, going over the main code blocks and the purpose of each method also demoing the execution of the code. YouTube links and links to google drive or any other media are not acceptable, the actual recording file must be submitted.
7. Any submission without an accompanying video will lose 25% of the grade.
8. Any submission without an accompanying Analysis report will lose 20% of the grade.

Assignment Pre-requisites:

1. Dialogflow
2. Tensorflow
3. Keras
4. Json
5. Pickle
6. Sample files with this assignment

Assignment Exercises

Exercise #1: Chatbot using API – group exercise **(25%)**

IMPORTANT. Do this exercise only if **one** of the following apply to you:

1. You were absent in Lab 5 (Date 10-2-2022).
2. You attend part of Lab 5 (not the whole session).
3. You did not show up when the group presented their chatbot at the end of Lab 5.

Exercise requirements:

In this exercise, you will create a goal-based chatbot using the Dialogflow API. The steps are detailed in Chapter #6 of your textbook, with an example of a Pizza ordering chatbot (refer to Lab 5 for more details).

Get approval from the professor on the topic of your goal-based agent. Then follow the steps in your textbook.

1. You should define at least one intent for the chatbot.
2. Provide a link to the chatbot you created.
3. Provide a screenshot showing the interaction with the bot you created. Name the screenshot "Firstname_bot_group#" where firstname is your first name and # is your group#.

Exercise #2: Simple chatbot using Deep learning **(75%)**

Scenario: you have been tasked to create a simple chatbot for your company to address some service issues in the areas of opening accounts and identifying complaints. You don't have any training data but was provided with a Json file with a sample of expected intents, patterns (phrases) & responses. The file only contains two intents as per the attached file sample_intents.json:

Exercise requirements:

1. Download the sample_intents.json attached to this assignment and rename it to "firstname_intents.json", where firstname is your first name.
2. If your first name starts with A-M then consider you are tasked to address the account opening problem; otherwise, you are tasked to address the "receive complaints" problem.

3. Prepare and load your data: 30%
   a. Add to the intents file, either manually or through a program, six new intents. Per intent, add three expected patterns (utterance) based on what you think the user might ask and three to five expected responses. Think carefully.
   b. Record the details of the intents, utterances, and responses in your analysis report with a clarification of why you chose each.
   c. Write code to read your json file and store the elements into a group of lists. (Hint: you will need four lists)
4. Pre-processing: Apply the following: 20%
   a. Encode the list of intents; these will be the classes for your model in total 8 classes. (hint: consider sklearn LabelEncoder)

   b. Use Keras tokenizer to tokenize the patterns, then carryout the necessary pre-processing steps that you think are required to prepare the data for a deep learning training on the intent patterns in order to identify intents. (Hint: consider fit_on_texts , word_index, texts_to_sequences, pad_sequences)

5. Deep learning training :
   a. Train a deep learning model to learn the intents from the pattern inputs, with the following parameters: 20%
      i. An embedding layer as input. Limit the embedding layer dimensions to 16, limit the vocabulary size to 1000 and limit the input_length to 20.
      ii. A pooling layer uses global average.
      iii. Two dense layers, each with 16 units and an activation function type "relu"
      iv. A final dense layer with units equal to the total number of classes i.e. # of total intents. Use softmax as the activation function.
      v. Compile your model and specify the following parameters:
         1. Loss function: sparse_categorical_crossentropy
         2. Optimizer: adam
         3. Metrics: accuracy
      vi. Print out the model summary, take a screenshot and add it to your analysis report.
   b. Fit the data to your model, try training with 300 epochs and then 500 epochs.
   c. Record the accuracy in your analysis report for each run and write some conclusions.
6. Testing your bot. 30%
   a. Save your tokenizer to a folder on your hard disk. (hint check pickle)
   b. Save your encoder to a folder on the hard disk. (hint check pickle)
   c. Save your model to a folder on the hard disk. (hint check TensorFlow save model)
   d. Write code to test your bot. You will need to load the model and pre-processing (tokenizer, encoder) in this script in addition to the json file. The code should cater for the following:
      i. Receive input from the user. (hint use python input())
      ii. Pass the input to the loaded model after passing it through the tokenizer.
      iii. De-encode the result.
      iv. Check if the result matches any of the intents in your intents json file, then randomly choose any of the stored responses and print it out to the user.
      v. Continue accepting input from the user, and responding, until the user enters an "exit" text.
   e. In your written analysis report suggest ways to improve your bot.

Naming and Submission Rules:

1. You must name your submission according to the following rule: **YourFullname_COMP262_assignmentnumber**.Example: **AdamPerjouski_COMP262_assignment1**
2. Please add all the commands/instructions into a python script or Jupyter notebook.
3. Upload the submission file on e-Centennial using the Assignment link(s).
4. In total you should submit the following:
   a. For exercise #1: One screenshot & the link

b. For exercise #2:
   i. Two python scripts
   ii. One Json file
   iii. One screenshot
   iv. One analysis report
   v. One demonstration video

Rubric (applies to each exercise #2)

Evaluation criteria	**Not acceptable**	**Below Average**	**Average**	**Competent**	**Excellent**
	0% - 24%	**25%-49%**	**50-69%**	**70%-83%**	**84%-100%**
Requirements in exercises	Missing all requirements required	Some requirements are implemented.	Majority of requirements are implemented but some are malfunctioning.	Majority of requirements implemented.	All requirements are implemented Correctly.
Instruction/ Code Documentation on python script	No comments explaining code. Missing screenshots	Minor comments are implemented.	Some code is correctly commented.	Majority of code is correctly commented.	All code is correctly commented.
Written analysis Content	Missed all the key ideas; very shallow.	Shows some thinking and reasoning but most ideas are underdeveloped.	Indicates thinking and reasoning applied with original thought on a few ideas.	Indicates original thinking and develops ideas with sufficient and firm evidence.	Indicates synthesis of ideas, in-depth analysis and evidences original thought and support for the topic.
Written analysis report format and organization	Writing lacks logical organization. It shows no coherence and ideas lack unity. Serious errors. No transitions. Format is very messy.	Writing lacks logical organization. It shows some coherence but ideas lack unity. Serious errors. Format needs attention, some major errors.	Writing is coherent and logically organized. Some points remain misplaced. Format is neat but has some assembly errors.	Writing is coherent and logically organized with transitions used between ideas and paragraphs to create coherence. Overall unity of ideas is present. Format is neat and correctly assembled.	Writing shows high degree of attention to logic and reasoning of all points. Unity clearly leads the reader to the conclusion. Format is neat and correctly assembled with professional look.
Demonstration Video	Very weak no mention of the code changes. Execution of	Some parts of the code changes presented.	All code changes presented but without	All code changes presented with explanation, exceeding time	A comprehensive view of all code changes presented with explanation,

	code not demonstrated.	Execution of code partially demonstrated.	explanation why. Code demonstrated.	limit. Code demonstrated.	within time limit. Code demonstrated.

Demonstration Video Recording

Please record a short video (max 8 minutes) to explain/demonstrate your assignment solution. You may use the Windows 10 Game bar to do the recording:

1. Press the Windows key + G at the same time to open the Game Bar dialog.
2. Check the "Yes, this is a game" checkbox to load the Game Bar.
3. Click on the Start Recording button (or Win + Alt + R) to begin capturing the video.
4. Stop the recording by clicking on the red recording bar that will be on the top right of the program window.

(If it disappears on you, press Win + G again to bring the Game Bar back.)

You'll find your recorded video (MP4 file), under the Videos folder in a subfolder called Captures.

Or

You can use any other video recording package freely available.