COMP5046 Assignment 1 [20 marks]

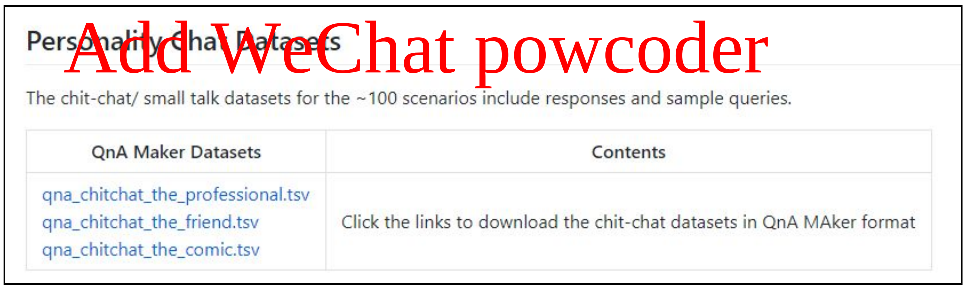
Basic Chatbot

Chatbots (Chat-oriented Conversational Agent) are designed to handle full conversations, mimicking the unstructured flow of a human to human conversation. In this assignment, you are to implement a basic (chit-chat) chatbot using Sequence to Sequence (seq2seq) model and Word Embeddings. The detailed information for each implementation step was specified in the following sections. *Note that lab exercises would be a good starting point for the assignment. The useful lab exercises are specified in each section.*

1. Data Preprocessing for Chatbot [3 Marks]

In this assignment, you are asked to use Microsoft BotBuilder Personality Chat Datasets, which includes three different personality chat (*professional, friend, comics*) datasets. With these datasets, you are to build our social chatbot and implement changing personality function *- refer to the Section 3*. In this Data Preprocessing section, you are required to implement the following functions:

- **Download all three datasets on Google Colab virtual server** (https://github.com/Microsoft/BotBuilder-PersonalityChat/tree/master/CSharp/Datasets).The three following datasets (qna_chitchat_the_professional.tsv, qna_chitchat_the_friend.tsv, and qna_chitchat_the_comic.tsv) should be downloaded on the Google Colab virtual server.



Personality Chat Datasets

The chit-chat/ small talk datasets for the ~100 scenarios include responses and sample queries.

QnA Maker Datasets	Contents
qna_chitchat_the_professional.tsv qna_chitchat_the_friend.tsv qna_chitchat_the_comic.tsv	Click the links to download the chit-chat datasets in QnA MAker format

Figure1. Microsoft BotBuilder Personality Chat Datasets

- **Preprocess data**: You are asked to pre-process the chatbot training data by integrating several text pre-processing techniques (tokenisation, removing numbers, converting to lowercase, removing stop words, stemming, etc.) - *Please refer to Lab 5*. You should justify the reason why you apply the specific preprocessing techniques. [*Justify your decision*]

2. Model Implementation [7 Marks]

In the 'Model Implementation' section, you are to implement two models, word embedding model and Sequence model. While the model is being trained, you are required to display the *Training Loss* and the *Number of Epochs*. You are free to choose hyperparameters (size of vector for embeddings, learning rate, etc.)

1) Word Embeddings [3 marks]

First, you are asked to build the word embeddings model for your chatbot as you will use word embedding (vector represents of words - such as word2vec-CBOW, word2vec-Skip gram, fastText) as input for seq2seq model. *Note that we used one-hot vector as input for seq2seq model in the Lab4.* In order to build the word embedding model, you are required to implement the following functions:

- **Download Data *for* Word Embeddings**: *This can be different from the data you used in the section 1 ('Data Preprocessing for Chatbot').* You can download and use any datasets to train the word embedding model: i.e. Microsoft BotBuilder chat datasets, Cornell Movie Dialog Corpus, etc. [*Justify your decision*]
- **Preprocess data for word embeddings:** You are to preprocess data for word embeddings - *refer to lab2 and lab3. This can be different from the preprocessing technique that you used in the section 1 ('Data Preprocessing for Chatbot')* [*Justify your decision*]
- **Build training model for word embeddings:** You are to build the training model for word embeddings. You are required to describe how hyperparameters (size of vector for embeddings, learning rate, etc.) Note that any word embeddings model (e.g. word2vec-CBOW, word2vec-Skip gram, fastText) can be applied. *- refer to Lab02 (Gensim) Lab03 (Tensorflow)* [*Justify your decision*]
- **Train model:** You can use *gensim package* or implement with *tensorflow.* While the model is being trained, you are required to display the *Training Loss* and the *Number of Epochs*.
- **Save model:** You are to save the trained word embedding model to your *Google Drive -refer to lab5*. *Note that your assignment 1 will not be marked if you modify the model after the submission.*
- **Load model:** You are to implement a function to load the model, saved in your *Google Drive*.

2) Sequence Modelling [4 marks]

Secondly, you are asked to build the Many-to-One (N to 1) Sequence model in order to train a chatbot. You must train three different individual sequence models (using three different personality datasets - *refer to section 1*) and those models should be individually applied. *Note that your chatbot users should be able to change the personality (trained model) of chatbot - refer to section 3 'Evaluation'.* You are required to implement the following functions:

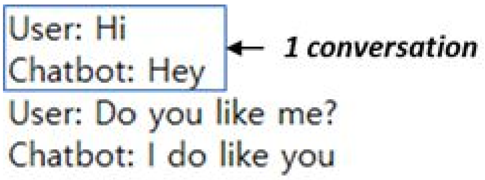
- **Apply/Import Word Embedding model:** You are to apply the trained word embedding model to the sequence model
- **Build training sequence model:** You are to build the Many-to-One (N to 1) sequence model -*refer to lab4*. You are required to describe how hyperparameters (the Number of Epochs, learning rate, etc.) were decided. [*Justify your decision*]

- **Train model:** While the model is being trained, you are required to display the *Training Loss* and the *Number of Epochs*.
- **Save model:** You are to save the trained sequence model to your *Google Drive* -*refer to lab5*. *Note that your assignment 1 will not be marked if you modify the model after the submission.*
- **Load model:** You are to implement a function to load the model, saved in your *Google Drive*.

3. Evaluation (Running chatbot) [4 Marks]

After completing all model training - *refer to the section 1 and 2*, you should apply the model to run the chatbot. *Users should be able to chat with the trained chatbot.* All chat log (chat history) needs to be printed on the console and saved in the ‘chat_log.txt’ file on your *Google Drive*. The following functions need to be implemented:

- **Start chat:** You are to implement the function to execute the chat. This function should include all required functions to load the trained chatbot (*download and load sequence model that you built in section 2*).
- **Change personality:** Users should be able to change chatbot personality during the conversation (runtime). The default personality should be “Professional”. The commands for personality change should be defined by you and implemented by using handcrafted rules (i.e. regular expression or exact string matching). The commands need to be written in the description section.
- **Save chat log:** All chat log (chat history) needs to be printed on the console and saved in the ‘chat_log.txt’ file on your *Google Drive*. The final chatbot requires to test more than more than 50 conversations. *The chatlog file should be submitted.* The following figure shows how the chat log has to be formatted.



User: Hi
Chatbot: Hey
User: Do you like me?
Chatbot: I do like you

1 conversation

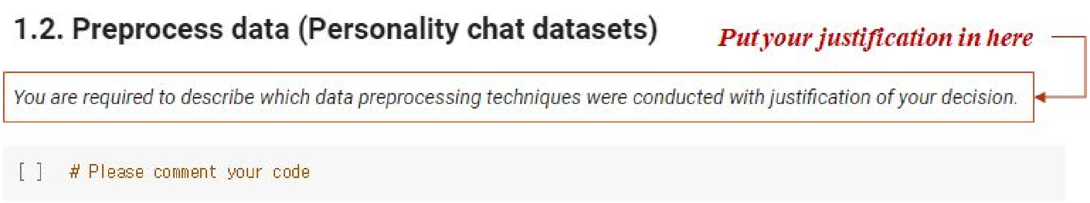
Figure2. Chat log format - example

- **End chatting:** You should define the ‘end conversation’ command, and the command should be implemented by using handcrafted rules. The commands need to be written in the description section.

4. Documentation [4 Marks]

In the section 1,2, and 3, you are required to describe justify any decisions you made for the final implementation. You can find the tag ‘[*Justify your decision*]’ for the point that you should justify the purpose of applying the specific technique/model.

For example, for section 1 ***(preprocess data),*** you need to describe which pre-processing techniques (removing numbers, converting to lowercase, removing stop words, stemming, etc.) were conducted and justify your decision (the purpose of choosing a specific pre-processing techniques, and benefit of using that technique or the integration of techniques for your chatbot) in ***your ipynb file***



1.2. Preprocess data (Personality chat datasets)

Put your justification in here

You are required to describe which data preprocessing techniques were conducted with justification of your decision.

```
[ ]  # Please comment your code
```

Figure3. The position of writing justification in ipynb file

5. Programming (coding) styles [2 Marks]

Your program needs to be easily readable and well commented. The followings are expected to be satisfied:

- **Readability**: Easy to read and maintain
- **Consistency & Naming**: Names are consistent in style
- **Coding Comments**: Comments clarify meaning where needed
- **Robustness**: Handles erroneous or unexpected input

Assignment 1 Submission Method

Submit a ipynb file that contains all above (1,2,3,4 and 5) contents.The ipynb template can be found in the Assignment 1 template. You also need to submit chatlog (txt file), word embedding model (zip file), and sequence model (zip file).

Due date: 5:00PM Monday 29 April 2019
Submission: Canvas Assignment 1 Submission Box
Submission Files:

- **ipynb file -** (file name: *your_unikey*_COMP5046_Ass1.ipynb)
- **txt file - chatlog** (file name: *your_unikey*_chat_log.txt)
- **zip file - word embedding model** (file name: *your_unikey*_embeddings.zip)
- **zip file - sequence model** (file name: *your_unikey*_sequence.zip)