Fall 2024 | DATA 255 | Homework -4
Deadline – 11.59 PM – 11/26/2024
20 Points

Problem 1: (1+6+1=8 points total) Use the training dataset from Analytics Vidhya to identify the sentiments of tweets. The training dataset is also available on Canvas. You should only use the training data. From this, you can split and create training, validation and test data. The highest performing model will receive a **2-point bonus.**

1. Build the sentiment analysis model:

   a. Implement text preprocessing steps. You may apply various preprocessing techniques based on your task needs, such as tokenization, removing stopwords, stripping HTML, converting to lowercase, and lemmatization/stemming. You may include other preprocessing steps if necessary.

   b. Utilize a combination of different numerical data representations and sequential models: Numerical Representation: TF-IDF, Word2Vec, and GloVe

   Sequential models: RNN, LSTM, GRU, and BiLSTM.

   Finally, provide a table that includes the results of all combinations.

   c. Minimum Expected Performance: Achieve an F1 score of at least 92% from one of the combinations.

Problem 2: (7 points) Use the text data from MOBY-DICK book for developing a simple language model that you can use it for text generation. Use minimum five chapters of the book. Develop a multiclass classification sequential model (e.g., GRU/LSTM/RNN) for this purpose. You must use at least 25 tokens as X features and the immediate next token as the Y feature. The generated text should not consist solely of identical words and must include a minimum of 50 words (tokens). The highest performing model will receive a **2-point bonus.**

1. Text Preprocessing (1 point):

   a. Tokenization

   b. Convert to lowercase

   c. Expand contractions

   d. Remove punctuation

   e. Apply lemmatization/stemming

2. Text Generation Model (6 points):

   a. (4 points) Develop the text generation model using Keras word embeddings.

b. (2 points) As a byproduct of training, you will obtain word embeddings for all words used in the model. Use cosine similarity to find the five words most similar to "whale"

Problem 3 (5 points): Developing a Naive RAG Model for Privacy Policy Retrieval - In this assignment, you will build a Naive Retrieval-Augmented Generation (RAG) model using the Google Privacy Policy document (uploaded on Canvas). The goal is to demonstrate the effectiveness of Naive RAG versus non-RAG models in answering specific policy questions.

Use 3-sentence chunks with a sliding window of 1 sentence to maintain overlapping context. Print the first five chunks. For queries and chunks encoding, use Sentence-BERT (a BERT-based model optimized for document and sentence embeddings, such as all-MiniLM-L6-v2 or distilbert-base-nli-stsb-mean-tokens) from Sentence Transformers.

For answer generation, use minimum **two LLM models** (e.g., GPT-2) to observe the difference between retrieval-augmented and non-retrieval models. Use these four questions:

1. What data does Google collect from users?
2. When does Google share user data externally?
3. What privacy controls are available to users?
4. How does Google manage cookies and tracking?

Finally, make a comparative discussion based on the output of these questions using different RAGs and Non-RAG models.

Useful link-

MOBY-DICK book - https://cs.brown.edu/courses/csci0931/2014-spring/2-text_analysis/HW2-2/MobyDick.txt

Sentiment analysis data - https://datahack.analyticsvidhya.com/contest/linguipedia-codefest-natural-language-processing-1/

Google Privacy Policy doc - https://static.googleusercontent.com/media/www.google.com/en//intl/en/policies/privacy/google_privacy_policy_en.pdf

Gensim- https://radimrehurek.com/gensim/models/word2vec.html

You are required to submit:

1. An MS/PDF/Scanned document:
    a. Include all the steps of your calculations.
    b. Attach screenshots of the code output.
    c. Include the summary of the model
    d. Include a Table - Mention all the hyperparameters you selected: activation function in hidden layer and output layer, weight initializer, number of hidden layers, neurons in hidden layers, loss function, optimizer, number of epochs, batch size, learning rate, evaluation metric
2. Source code:
    a. Python (Jupyter Notebook)
    b. Ensure it is well-organized with comments and proper indentation.

- Failure to submit the source code will result in a deduction of 5 points.
- Format your filenames as follows: "your_last_name_HW1.pdf" for the document and "your_last_name_HW1_source_code.ipynb" for the source code.
- Before submitting the source code, please double-check that it runs without any errors.
- Must submit the files separately.
- Do not compress into a zip file.
- HW submitted more than 24 hours late will not be accepted for credit.