Look, Ask, Explain: The VQA Challenge

Week 2 Assignment

Hey all, we hope you enjoyed learning this week and gained valuable insights. Let's apply your knowledge to the following problems!

Problem 1

Use the Twitter US Airline Sentiment dataset (available on Kaggle), which contains tweets labelled with the sentiment of the user toward airlines (positive, negative, or neutral). The data is stored in a Pandas DataFrame with columns such as *airline sentiment*(target) and text (tweet content). Perform the following tasks:

- Preprocess each tweet using the following steps:
  - Convert the text to lowercase.
  - Remove URLs, mentions (e.g., @username), hashtags, and punctuation.
  - Expand common contractions (e.g., "don't" $\rightarrow$ "do not")
  - Lemmatise the words (use NLTK).
  - Optionally remove emojis and special symbols
- Load the pre-trained Google News Word2Vec model using *gensim*.
- Convert each tweet into a fixed-length vector by averaging the Word2Vec word vectors for all words in the tweet. Ignore words not found in the embeddings.
- Split the dataset into training (80%) and testing (20%) sets using train-test split of the sklearn library.
- Train a Multiclass Logistic Regression classifier on the vectorised training data and report the accuracy on the test set.
- Write a Python function *predict tweet sentiment(model, w2v model, tweet)* that takes the trained classifier, the Word2Vec model, and a single tweet (string), and returns the predicted sentiment (positive, negative, or neutral).

Problem 2: Creating a Machine Learning Pipeline with Hugging Face

Your task is to design and implement a machine learning pipeline using Hugging Face's libraries to perform sentiment analysis on a text dataset. Use the transformers library to fine-tune a pre-trained BERT model on the IMDb dataset available through Hugging Face's datasets library. Your pipeline should include the following steps:

- Load the IMDb dataset using the datasets library.
- Preprocess the dataset, including tokenisation using the appropriate tokeniser for bert-base-uncased.
- Fine-tune the BERT model for sentiment analysis (binary classification: positive or negative).
- evaluate the model's performance using accuracy and F1-score metrics.
- Save the fine-tuned model and demonstrate how to load it for inference on a sample text input.

Provide the complete Python code for the pipeline, including necessary im- ports and comments explaining each step. Additionally, include a brief written explanation (150–200 words) describing the pipeline, its components, and the rationale behind your design choices. Discuss any challenges you anticipate (e.g., computational requirements, data preprocessing) and how you would address them. Submit your code as a Python script and the explanation as a separate section in your report.

- Ensure your code is executable and includes error handling where appropriate.
- Provide a link to the fine-tuned model if uploaded to the Hugging Face Model Hub.

Resources

- Hugging Face Datasets: https://huggingface.co/docs/datasets/
- Hugging Face Transformers: https://huggingface.co/docs/transformers
- IMDb Dataset: https://huggingface.co/imdb/datasets

Happy Learning :)