CS410 Text Information Systems – Team: Fan & Jain Rhymes

Progress report

Task completed:

Below are the tasks we have completed so far:

1. Solution Architecture Design & Approach Finalization:
   Team planned to use several state-of-art architectural styles to experiment & validate the results & compare different approaches results. Approach is to start from conventional styles and gradually progress to use advance methodologies.
   a. Conventional ML methodologies
   b. Deep Learning based implementations
   c. Attention & Transformers based implementations.

2. Environment setup
   a. Analyzed different environment viz. local desktop, Google Colab and Cloud to setup experimentation playground. Cost effectiveness & high processing needs were the key parameters considered.
   b. Colab Pro environment was preferred over others which enabled us using High Memory & GPU/TPU based processing for Deep Learning based implementations.

3. ML Pipeline setup:
   a. Data Import: Training & Test data was imported to google drive & authentication setup was done to access it.
   b. Data Preprocessing: Data clean step is performed to address words spelling error, repetition, signs & emoji.
   c. Feature Engineering: Several features were constructed to support solution approach as multi sentence sequence & single document classification problem.
   d. Model Training/Fine Tuning: Select, train, and evaluate the model. For pre-trained models fine tuning step was performed considering different solution classification styles.
   e. Prediction: Output the predictions.

4. ML Models: Below are several modelling strategies which team has evaluated for given Classification Problem.
   a. TF-IDF + dimensionality reduction (via SVD) + Tree (Random forest).
   b. General LSTM model (with one or multiple LSTM layers followed by fully connected layers).
   c. Transformer (attention-based) models leveraging hugging face pre-trained models: Roberta, BERT, XLM, based on which we fine-tune for our task.

5. Observations:
   We have compared all the approaches & results are per our expectation as below:
   - Conventional ML (TFIDF/SVD/Random Forest) based implementation was not able to beat the baseline score.
   - Deep Learning (LSTM based) implementation <WENXI TO CONFIRM>
   - Transformer based approach:
     - Bert base: Just at par with Baseline results but ranked intermediate on leader board.

- Roberta base: Performed very well with highest accuracy, precision & F1 values.

Task pending:

Below are the tasks we plan to complete before the final submission:

1. We are planning to further explore other models for our task, that includes: GPT-2, GPT-3.
2. We are planning to complete the documentation of our pipeline to be prepared for the final submission.

Challenges faced:

Below are the challenges we faced:

1. Noises in the data:
   The input data appears to have a lot of noises – words spelling error, repetition, words/signs/emoji that appear to occur very few times. In addition, there are many words/signs in test sets that are not included in the training set. These create challenges in extracting useful information out from the inputs and challenges in generalizing the model to the data not included in the training process.

2. The way to leverage context/response to engineer effective input features:
   Currently we use the context/response by inputting them as two separate features or concatenating as one (and/or reversing orders). Although we leveraged the transformer models that generate attention mechanism(both on words and position of words), we have not yet been able to explore a way ourselves to engineer features that might be more effective as inputs than pure words and sentences. The example of such inputs could be sentiment of sentences, the hierarchical attention from character-level to sentence level.

3. Hyper-parameter tuning
   We found challenges in hyper-parameter tuning, especially when the parameter space is large. We used 'Trainer' (utility from transformers) that helped automate the parameter search, but still the parameter space is large, and getting the best model may take time.

4. Large Model
   We are unable to tune large transformer models like ROBERTA_LARGE & BERT_LARGE models due to GPU memory issues.

5. Select the performance criteria
   We found challenges in selecting models that generates well on test data – although we used validation set in addition to test set (which sets are separated out from training data), the model that test well (good F1) on validation and test set may not generalize well to the other data not seen in the training process.
   Instead of just using F1 score, the one datapoint, as the criteria to judge the performance, the below steps were performed that achieved better performance criteria (which gives better picture on the performance of model):
   a. Evaluate the distribution of scores on multiple batches of the validation set, to analyze the level and consistency of the performance.

b. In addition to evaluate precision/recall/f1 the level of performance, we could also evaluate from a different perspective, such as analyzing the correlation between prediction and ground truth.