Clement Canel, Ratnakshi Gore & Dnyaneshwari Rakshe

4/25/2025

CSCI 5622

Homework #4 Write Up

A. In this project we extracted multiple types of language features. To meet the requirement of spanning different feature categories, both syntactic vectorizers and semantic features were extracted. First, at the syntactic level, we create unigram, bigram, and trigram representations of the cleaned transcript text. CountVectorizer allowed the model to numerically represent the frequency of specific words or short word combinations, capturing simple yet powerful patterns. To complement this, we also extracted TF-IDF (Term Frequency-Inverse Document Frequency) features. TF-IDF provided a nuanced weighting scheme, highlighting words that were frequent in one response but rare across the dataset.

Beyond syntactic features, we incorporated semantic-level features to enrich the understanding of deeper meaning and emotional context in responses. This included sentiment analysis (polarity and subjectivity scores) allowing the model to detect whether answers leaned positive, negative, neutral, or subjective. Further, topic distribution features were generated using topic modeling, assigning each answer to a topic cluster and recording its topic probability. Lastly, we computed semantic similarity between questions and answers, quantifying how directly participants answered questions.

B. To address the task of classifying speaker identity, we implemented both a tree-based and a deep learning model. For the tree-based model, we selected XGBoost (Extreme Gradient Boosting) due to its proven effectiveness in handling structured and sparse data like the n-gram features extracted from the transcripts. For the deep learning approach, we developed a Feedforward Neural Network (FNN) to capture more complex and nonlinear patterns in the feature space. We utilized a filter-based feature selection method using the chi-squared (χ^2) test. This statistical method allowed us to rank features based on their relevance to the speaker identity task.

We employed a 5-fold Cross-Validation strategy, where the data was randomly split while ensuring that samples from the same participant could be present in both the training and testing sets. For each fold, both models were trained independently, and two evaluation metrics were reported: Simple Accuracy and Balanced Accuracy. Our experiments showed that the XGBoost model, when trained on n-gram features (unigrams, bigrams, trigrams) achieved an average accuracy between 0.76 and 0.763 depending on the n-gram range. The Feedforward Neural Network, trained on the top 200 TF-IDF selected features, performed even better, achieving an average accuracy of 0.783 and a balanced accuracy of 0.737 over 5 folds. These results indicate that the deep learning model was able to leverage the richer linguistic patterns captured by the TF-IDF vectors more effectively.

To enhance model interpretability, we further employed SHAP (SHapley Additive exPlanations) analysis on the XGBoost model. SHAP helped us identify key n-grams that contributed most strongly to the speaker classification.

C. For the binary classification task distinguishing under-explained from succinct responses, we used Random Forest and a Neural Network with TF-IDF features, applying mutual information-based feature selection to identify the top k=100, 200, and 300 features. Data was split into 5 participant-independent folds to ensure no participant appared in both train and test sets, usinonly under-explained and succinct responses. RF slightly outperfrmed NN (A=0.7915 vs. 0.7915, BA≈0.5 for both), but low BA indicated struggles with the minority under-explained class. Feaure overlap between speaker identity and classification tasks increased with k (30 to 65), risking privacy leakage, while per-speaker accuacy varied widely (e.g., P007: 0.4, P002: 0.6667), reflecting stylistic diffrences. Suggest class imbalance and speaker-specific biases require privacy-focused feature filtering and imbalance correction techniques.

D. While implementing the binary classification task to classify over-explained and comprehensive responses, we used Random Forest and a Neural Network with TF-IDF features. We applied mutual information-based feature selection to identify the top k values = 200, 300, and 400 features. We used 5 participant-independent folds to split the data so that no participant appeared in both train and test sets and is filtered properly for over-explained and comprehensive responses. Based on the results, we observed that the random forest algorithm slightly outperformed Neural Network model. We received accuracy of approximately 0.76 for both the models and balanced accuracy of approximately 0.5 for both the models. The low value of balanced accuracy highlighted the biasness within the data. Based on these observations. We can say that the class imbalance and the biasness in the data will require privacy-focused feature filtering and imbalance correction techniques.

E. We removed the top 100 speaker-informative TF-IDF features using mual information, applied Fairlearn's ExponentiatedGradient (DemographicParity), and classified under-explained vs. succinct responses with Random Forest (RF), RF_Fairlearn, and Neural Network (NN) across k=100 200, 300 features in 5 participant-indeendent folds. RF_Fairlearn outperfomed RF and NN (BA=0.5126 at k=100 vs. 0.4108 and 0.5), but all strugled with under-explained samples (A=0.6549–0.915). Feature overlap with speaker identity rse from 33 (k=100) to 77 (k=300), risking privy, while per-speaker accuracy varied (P006: 0.0, P002: 0.6667).

F. The minGPT is basically a lightweight version of GPT, written in PyTorch. It's mainly built for learning purposes, so we made some changes to use it for classification. We replaced its original output layer with a new one that can predict class labels, used a character-based tokenizer to match our data, and added a way to combine token outputs into a single sentence representation. We also trained it with a few labeled examples, which helped the model learn how to classify text using its existing language understanding. We used confusion matrix to evaluate the results. The confusion matrix showed the classification performance for four categories: "Under-explained," "Succinct," "Comprehensive," and "Over-explained." It helps us in understanding that the model performs well in predicting "Comprehensive" responses, with 14 correct classifications (true positives) and minimal confusion with other categories. However, some overlap exists between "Succinct" and "Comprehensive," where "Succinct" is misclassified as "Comprehensive" 11 times. Additionally, "Over-explained" responses are consistently misclassified as "Comprehensive," indicating challenges in distinguishing between these two categories. Overall, the model has a bias toward predicting "Comprehensive," potentially overshadowing other classifications. Then we also used it with a GPT2 tokenizer for few-shot classification of under-explained vs. succinct

responses, using four example prompts and testing on 50 samples (rows 50–100). It scored low (accuracy 0.63, balanced accuracy 0.53), behind Random Forest (0.82) and Fairlearn RF (0.81) from Parts C and E, due to a small dataset (130 samples) and text truncation. Speaker accuracy varied a lot (P006: 0.0, P002: 0.6667), with short replies (15 wors) often misclassified compared to succinct ones (28 words). Finetuning or a stronger model, plus fairness tweaks like Fairlearn, could boost results and fix speaker biases. In short, while working on this experiment we observed various limitations of minGPT like it uses smaller architectures as compared to LLMs. It lacks advanced attention mechanisms required for context understanding and it also doesn't get updated with new information.