COMS 6998: Advanced Topics in Spoken Language Processing
Spring 2025

Homework 2 - Dialog Systems and Acts (100 points)

In this assignment, you will be working with dialogue acts and systems across two tasks.

General submission details:
Submit a zipped folder <uni.zip> containing two folders: “task1” and “task2”. Submit to Gradescope.

Task 0: GPT-use affirmation

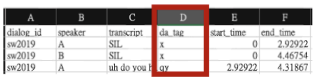
Please confirm your compliance by typing the following statement at the top of your report: **'I confirm that I have not used any GPT-generated responses for any part of this assignment.’**

Task 1: Dialogue Act Recognition (85 points)

For this task, you will work on the problem of Dialogue Act Recognition (DAR). A dialogue act is an utterance in a dialogue that serves a particular function. DAR is fundamental for spoken language understanding and is an important component of spoken dialogue systems.

Submission details for folder “task1” (5 points):

1. Feature Extraction and analysis script (i.e., **features.ipynb**, code for task 1-1 and task 1-2)
2. Classification script(s) (i.e., **classification.ipynb,** code for task 1-3)
3. CSV for each of your extracted feature sets (the features you used for your classification model), with feature names in header.
   (i.e., speech_features_train.csv, speech_features_valid.csv, speech_features_test.csv, text_features_train.csv, text_features_valid.csv, and text_features_test.csv)
4. **test_{UNI}_{speech, text, multi}.csv with your prediction results.**
   Note:
   - modify the file name of test.csv to test_UNI_{speech, text, multi}.csv (e.g. test_ab1234_speech.csv, test_ab1234_text.csv, test_ab1234_multi.csv)
   - fill in the “da_tag” column of the test.csv file with your model predictions

	released	submission
filename	test.csv	test_ab1234_speech.csv
csv		
	The “da_tag” of test data won’t be released.	Fill in the “da_tag” column in test_{UNI}_speech.csv with predictions of your speech model.

-

5. **PDF** write up file with your responses

([UNI]_task1_responses.pdf)

6. Documentation/README for the code
   (e.g., README.txt. README should include packages you used (i.e., requirements) and what we need to know to run your code.)

Note:

- Please retain your outputs in the IPython notebooks.

- All written responses (including the statistics and images) must be included in your report. Answers provided directly in the IPython notebook or references to answers within it will not be considered.

- Please DO NOT submit the wav files.
- Please DO NOT combine features.ipynb and classification.ipynb into one file.
- If your extracted file is too large to upload to Gradescope, include a link to your Google Drive where the features are saved in your responses file.

Data:

For this task, we make use of the Switchboard Dialog Acts corpus. This is a corpus of telephone speech annotated with dialog act tags. The disfluency annotation of the corpus is described here:
https://www.cs.brandeis.edu/~cs140b/CS140b_docs/DysfluencyGuide.pdf

We have split the data into train/valid/test sets for you and have also provided the corresponding audio at [Hw2 data].

- **Train csv file**: training set. Use the data in the training set to train your model.
- **Valid csv file**: validation set. Use the validation set for performance analysis and error analysis.
- **Test csv file**: test set. We will evaluate your model performance based on the test set performance. The label of the test set won't be released. If your model performs well on the validation set, it usually will also perform well on the test set. However, if you overfit your model on the validation data, performance on the test set may drop greatly.
- **Wav file**

1. **Feature Extraction (20 points)**

   Extract **two** feature sets that you feel would be useful for the DAR problem. One feature set should be text-based, and the other feature set should be speech-based. Save text-based and speech-based feature sets as text_features_{train, valid, test}.csv and speech_features_{train, valid, test}.csv, respectively.

   a. Describe your custom feature sets (text-based features and speech-based features), the reasoning behind choosing them and the techniques used to extract them.

   Note:

   - Some resources (see the resources link in the syllabus): NLTK, Stanford NLP toolkit, Praat / Parselmouth, openSMILE, LIWC
   - LIWC features are provided for you in train and test csv files, all columns after end_time are LIWC features. You can decide whether to: not use at all, use

a subset, or use all LIWC features.

- Feature files should have a similar format to the “train.csv” and “test.csv”. The first four columns should be “dialog_id,” “speaker,” “da_tag,” “start_time”, “end_time.” as the original “train.csv” and “test.csv.”
- ***We encourage you to explore more features and not be limited to LIWC and what you learned from HW1***.** You can use any toolkits for feature extraction. Describe the toolkits you used and explain the reasons behind your choices in your report.

2. **Feature Analysis (20 points)**

a. For each custom feature set (text-feature set and speech feature set), formulate and test a hypothesis about the features (visually or statistically). Observe if the results are in accordance with your hypothesis or not. Give an explanation about your thinking behind the observed behavior.

For example, testing whether the LIWC feature “Insight”, which is associated with words such as “think” and “know”, or the bigram “I think” are useful in predicting the dialogue act “Statement-opinion”. This hypothesis could be tested by plotting average values of the LIWC “insight” features or “I think” bigram for the top 10 dialogue acts.

3. **Classification and Error Analysis (40 points)**

Using the feature sets, train machine learning classifiers to identify/predict **the 10 most frequent dialogue acts** (**Use the top 10 from section 3** here).

Note:

- You can decide whether to use the training samples that are not labeled as one of the top 10 dialogue act tags.

a. Model training. Train 3 models: (1) speech features only (2) text features only (3) speech + text features.

a-1. describe the model(s) you used.

Note: - You can use any model, including neural networks.

b. Performance analysis.

b-1. Report the performance on validation set:

Model	Accuracy	F1
Speech		
Text		
Speech + Text		

Note:

- Use average='**macro**' to calculate the F1 score: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html
- You only need to report the best model for each feature set. (i.e., you only need to report the performance of ONE model of speech, text, and speech+text.)

b-2. Which model performs the best (i.e, speech, text, or speech+text model)? Why do you think it performs the best?

c. Error analysis (on validation set). For your best model (i.e, speech, text, or speech+text model):

c-1. Show the confusion matrix, including two normalized and one original. (Original: raw confusion matrix without normalization. Two normalized: normalized confusion matrix over the true (rows) and predicted (columns) conditions.)

c-2. Which class(es) were easiest to predict? Why do you think they were easy?

c-3. Which were the most difficult? Why do you think they were difficult?

c-4. What are easily confused classes? Why do you think your classifier made these errors?

c-5. Based on this analysis, what ideas do you have to further improve your classifier/model?

Note:

- The class with the highest/lowest accuracy or F1 score is considered the easiest/most difficult. Easily confused classes are two classes that are frequently mistaken for one another, and you can identify them using a confusion matrix.

d. Submit predictions for the Three test sets **i.e., test_{UNI}_{speech, text, multi}.csv**.

- All samples in the test set belong to the top 10 dialog acts.
- Try your best for the model performance, but do not overfit your model. **Your grade in this part will depend on the model's performance. Submissions that demonstrate strong performance relative to the class will receive more points.**
- **We will evaluate your model performance based on the test set performance rather than the validation set performance**. If you didn't overfit your model, the performance for the validation set and test set should be close.
- The performance analysis above is for you to evaluate your own model. Your grade for the model performance section will be automatically graded, so ensure your files are in the correct format. Keep the 'dialog_id', 'speaker', 'start_time', and 'end_time' columns in the provided test CSV and fill in the 'da_tag' with your predicted labels. You can use the ***evaluate_performance.ipynb*** and validation set to check your submission format.

Task 2: Analyzing Spoken Dialogue Systems (15 points)

For this task, you will be working with popular spoken dialogue systems (sometimes

also referred to as “voice assistants” or “conversational assistants/agents”).

Submission details (for folder “task2”):
PDF write up file with your responses (i.e., ([UNI]_task2_responses.pdf)

Ask any two of Google Assistant, Alexa, Cortana or Siri (or any other voice assistant!) for restaurant suggestions given a particular cuisine and neighborhood. Then get the phone number and restaurant hours. Use the same queries for both systems in order to directly compare the performance.

1. Describe the experience that you had with each of the systems.

    a. What query or queries did you use, and how did the system respond?
    b. Was the system able to complete the task with the desired result and was the dialogue efficient?
    c. Describe errors made by the system and how the system or user recovered from them. If no errors occurred, suggest areas where the system could be improved.
    d. Which system did you prefer using, and why?