Homework 2 - Dialog Systems and Acts

I confirm that I have not used any GPT-generated responses
for any part of this assignment.

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.

Text-based Feature

For the text-based modality, I designed a hybrid feature set that combines:

1. Structural features
2. Part-of-speech distribution features
3. LIWC-style psycholinguistic features (provided with the dataset)
4. TF-IDF lexical features (unigram + bigram)
5. Contextual Embeddings from DistilBERT

This combined representation aims to capture linguistic style and lexical content, which are both strongly associated with dialogue act categories.

(A) Structural Features

I extracted several interpretable surface-level structural features from each transcript, including:

- Character length (len_chars)
- Token length (len_tokens)
- Punctuation count (num_punct)
- Whether the utterance ends with a question mark (ends_question)

Reason:
These features capture global structural properties of the utterance.
Many dialogue acts, such as Questions, tend to have short lengths and end with “?”
Acts like Statement-opinion or Statement-nonopinion are often longer and contain more complex punctuation
Backchannels (e.g., “uh-huh”, “yeah”) are very short and have very low token counts
Therefore, structural features are directly diagnostic of dialogue-act categories.

Extraction Technique：
All structural features were computed using simple tokenization via nltk.word_tokenize followed by direct string manipulations.
This ensures fast and transparent feature extraction with no risk of overfitting.

(B) POS Distribution Features

For each utterance, I computed the proportion of major POS categories:

* pos_prop_NOUN
* pos_prop_VERB
* pos_prop_ADJ
* pos_prop_ADV

Reason:
The distribution of parts of speech reflects the functional role of the utterance:
Questions typically have more verbs and adverbs (“how”, “what”, “do you”). Opinions tend to include many adjectives and adverbs (“really”, “pretty”, “good”, “bad”). Statements often contain more nouns, reflecting topic introduction. Backchannels and acknowledgements contain few POS items overall
By converting counts into proportions, POS features become independent of utterance length.

Extraction Technique:
POS tagging was implemented using:

```python
from nltk import word_tokenize, pos_tag
```

(C) LIWC-style Psycholinguistic Features

The dataset already includes a set of LIWC-derived counts for each utterance:

Examples include: pronoun, auxverb, negemo, insight, tentat, differ, work, money, social, family, posemo, anger, sad, etc.

Reason:
LIWC features capture psychological intent, social orientation, and cognitive state—all of which are strongly related to dialog acts.

For examples, Utterances containing “I think”, “I guess”, “I believe” often indicate Statement-opinion. High negation or negative emotion may correspond to Disagreement. Frequent pronouns may correlate with Personal topics or Opinions. High insight or cogproc loadings indicate reasoning or explanation acts. These features help capture interpretive, semantic cues beyond raw text.

Extraction Technique:

These features were directly read from the CSV and merged with structural + TF-IDF features.

(D) TF-IDF lexical features (Unigram + Bigram)

I trained a TF-IDF vectorizer on the training transcripts:

* n-gram range: (1, 2)
* max features: 2000
* lowercasing + unicode normalization enabled

Reason:
TF-IDF features provide direct lexical cues.
Unlike LIWC, TF-IDF captures specific content, such as “do you”, “are you” indicate very strong indicators of questions. “I think”, “I mean” means strong indicators of opinions. Topic-specific n-grams are useful for statements. Filler words (“uh”, “um”) are often correspond to backchannels or disfluencies
TF-IDF provides a high-dimensional sparse representation that significantly boosts predictive power, especially for linear models.

Extraction Technique:

```python
vectorizer = TfidfVectorizer(...
```

(E) Contextual Embeddings from DistilBERT

Description:
To complement traditional lexical and psycholinguistic features, I additionally extracted 768-dimensional contextual embeddings from DistilBERT. For each utterance, I fed the text into the pretrained DistilBERT model and performed masked mean pooling over token hidden states to obtain a single sentence-level embedding.

Reason:
Unlike TF-IDF, which is purely frequency-based, and LIWC, which captures interpretable psychological categories, BERT-style embeddings encode contextualized semantics — words are represented differently depending on their surrounding words. This allows the model to distinguish subtle intent differences such as “I think” vs. “Do you think”, or “Yeah, right” (sarcastic agreement) vs. “Yeah, that’s correct” (affirmative).
By combining these contextual vectors with LIWC and TF-IDF features, we can jointly model surface form, psychological signals, and deep semantic meaning, leading to a more comprehensive representation of dialog acts.

Extraction Technique:
I used the pretrained model "distilbert-base-uncased" from Hugging Face Transformers. Each utterance was tokenized (max len = 96), passed through the encoder on MPS, and pooled to a fixed-length vector. These embeddings were then concatenated with the structural, POS, LIWC, and TF-IDF features to form the final feature matrix before model training.

Total text feature dimension: 2854

This hybrid approach combines interpretable features with powerful lexical statistics, which together provide a rich linguistic representation well-suited for dialogue act classification.

Speech-based Feature

For the speech modality, I extracted a compact but informative set of 29 acoustic-prosodic features, designed to capture properties that are strongly linked to conversational structure and therefore highly relevant to dialogue act recognition (DAR). These features include pitch, intensity, jitter, shimmer, harmonicity (HNR), MFCCs and Self-Supervised Acoustic Embeddings computed on each time-aligned audio segment.

(A) Pitch-related Features
Features: pitch_min, pitch_max, pitch_mean, pitch_sd

Reason:
Pitch is one of the most important signals in conversational pragmatics:

- Yes/no questions tend to have higher final pitch or rising contours
- Wh-questions often begin with a high pitch
- Backchannels (e.g., “uh-huh”) tend to have low pitch variance
- Statements typically exhibit mid-range, stable pitch patterns
- Emphasis or opinions may show higher pitch range and variability

By including min/max/mean/std, I capture both absolute pitch level and prosodic dynamics, which help differentiate question-like, expressive, or monotonic acts.

Technique Used: Pitch features were extracted using Praat autocorrelation method

(B) Intensity-related Features

Features: int_min, int_max, int_mean, int_sd

Reason:
Intensity reflects emphasis, engagement, and speech energy, which correlate with several dialog acts:

- Greetings, Acknowledgements, and Backchannels are generally short and low-intensity
- Opinions and Complaints tend to show higher average intensity
- Questions often show increased amplitude near phrase boundaries
- Commands or Corrections may be louder or more forceful

The combination of mean, max, and variability provides a robust representation of speaking style and emotional force.

Technique Used: Using Praat's intensity extraction

(C) Voice Perturbation Features (Jitter & Shimmer)

Features: jitter (local), shimmer (local)

Reason:
Jitter and shimmer measure micro-variations in voice frequency and amplitude, respectively. These features are classically used in speech pathology and emotion recognition, but they are also informative for conversational behaviors:
* High jitter/shimmer often occurs in hesitations, uncertainty, and self-corrections
* Backchannels like "uh", "um" have unstable voicing indicates higher jitter
* Confident statements usually have stable voicing indicates lower jitter/shimmer
Thus, jitter and shimmer provide cues for confidence, hesitation, and speaker affect, all relevant for DAR.

Technique Used: Praat's measures

(D) Harmonic-to-Noise Ratio (HNR)

Feature: hnr_mean

Reason:
HNR measures voice clarity and breathiness:
* Low HNR may indicate breathy, tense, or uncertain speech, which is common in questions, disfluencies, corrections
* High HNR corresponds to clear, stable phonation, which is typical of statements or confident opinions
Thus, HNR provides a compact measure of vocal quality differences across dialogue acts.

Technique Used: Praat's measures

(E) MFCCs

For every segment, I computed MFCC from 1 to 13 and took the mean value of each.

Reason:

MFCCs capture the short-term spectral envelope — the “timbre” of the speech — which reflects vowel quality, articulation clarity, phoneme distribution and speaking style.

These cues differ systematically across DA types. For example:

* Backchannels / filled pauses (“uh-huh”, “mm”) show more low-frequency energy
* Questions show more high-frequency shift due to rising intonation
* Statements show flatter MFCC contours

MFCCs are widely used in speech recognition and emotion recognition, making them a robust addition to DAR.

Technique Used: Librosa

(F) Self-Supervised Acoustic Embeddings (WavLM-based)

Description:

In addition to the hand-crafted acoustic-prosodic features, I extract 768-dimensional contextual acoustic embeddings from the pretrained WavLM-Base+ model.

WavLM is a self-supervised transformer trained on large-scale unlabeled speech data, learning to represent high-level prosodic and phonetic information such as tone, speaker traits, and conversational cues.

Reason:

Traditional low-level features (MFCC, jitter, shimmer) capture local spectral statistics but fail to encode long-range context or semantic prosody. WavLM embeddings, by contrast, provide context-aware representations that model intonation patterns, speaking style, and discourse function.

For instance, rising pitch patterns at the end of an utterance may signal a Question, while elongated vowels and reduced energy often indicate Backchannels (e.g., “uh-huh”, “yeah”).

By combining these pretrained embeddings with hand-crafted features, we enable the classifier to jointly learn interpretable prosodic cues and high-level paralinguistic context.

Extraction Technique:

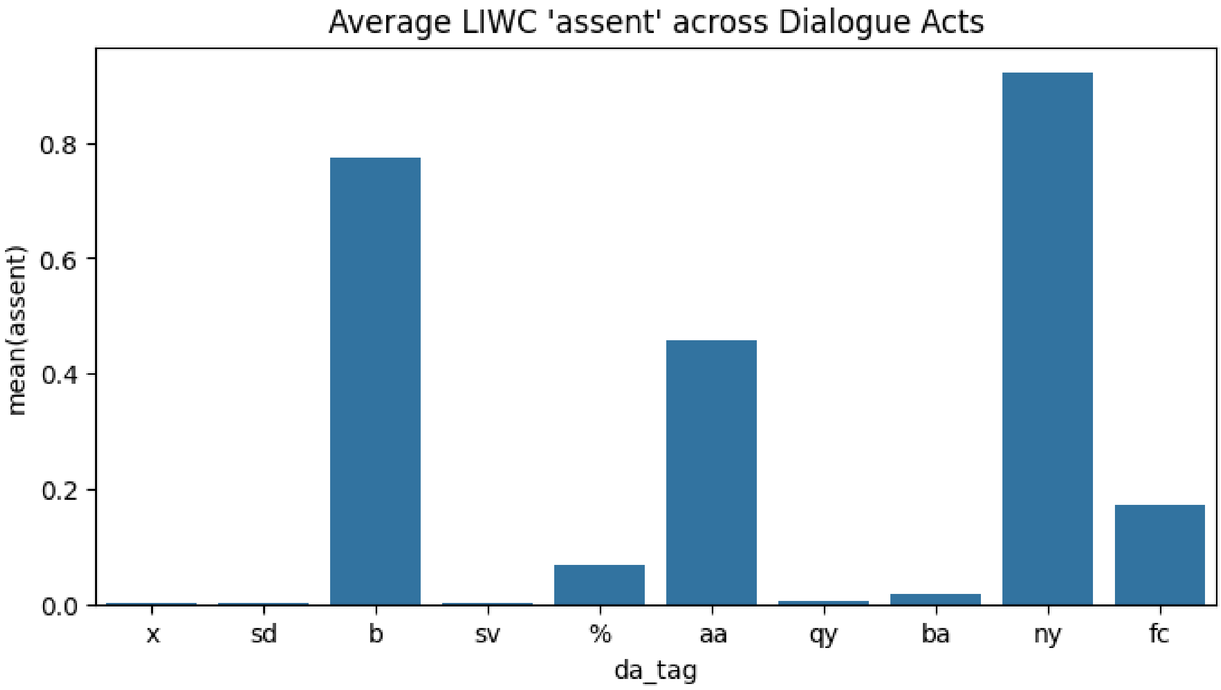
Each utterance segment is extracted according to the time boundaries in the dataset and passed to a pretrained microsoft/wavlm-base-plus model from Hugging Face Transformers. The mean-pooled output from the final hidden layer (dimension = 768) is used as the utterance-level embedding. These vectors are concatenated with the handcrafted acoustic features to form the final feature set.

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.

Text-based Hypothesis:

I hypothesized that Agree/Accept (aa) would show higher LIWC assent than other dialogue acts, because aa utterances typically contain “yes/yeah/right/okay”.



Code output:

```
Per-class mean(assent):
da_tag
x      0.000356
sd     0.001568
b      0.773436
sv     0.002087
%      0.067691
aa     0.455504
qy     0.004391
ba     0.016996
ny     0.920603
```

```
fc    0.171708
Name: assent, dtype: float64

Mann-Whitney U (aa > others): U=151738678.5, p=0.000e+00
```

Result:
Contrary to the hypothesis, when computing per-class means, the actual order was:

* ny (0.921) — highest
* b (0.773)
* aa (0.456)
* fc (0.172)
* (others ≪ 0.1)

A Mann–Whitney U test comparing aa vs all other classes yielded: U = 1.517e8, p ≈ 0

Although the p-value indicates strong differences between groups, the direction contradicts the original hypothesis.
aa does not exhibit the highest “assent” levels; instead, ny and b are higher.

Explanation:
1. Short-utterance length inflation
Many ny examples are extremely short responses (e.g., “yeah, no.”). In very short utterances, a single yeah dramatically increases the proportion of “assent” tokens, often near 1.0.

2. “Yeah, no” patterns
ny frequently begins with yeah but then negates the proposition (e.g., “yeah I don’t think so”). This causes assent to co-occur with negate, raising its average for ny.

3. Backchannels (b) naturally contain assent tokens
Backchannel responses (e.g., “yeah”, “right”, “uh-huh”) often contain exactly the words captured by LIWC “assent,” making high values expected.

4. Proportional LIWC features amplify certain DA types
Because “assent” is measured as proportion of tokens, shorter utterances inflate the value much more than longer agreement statements.

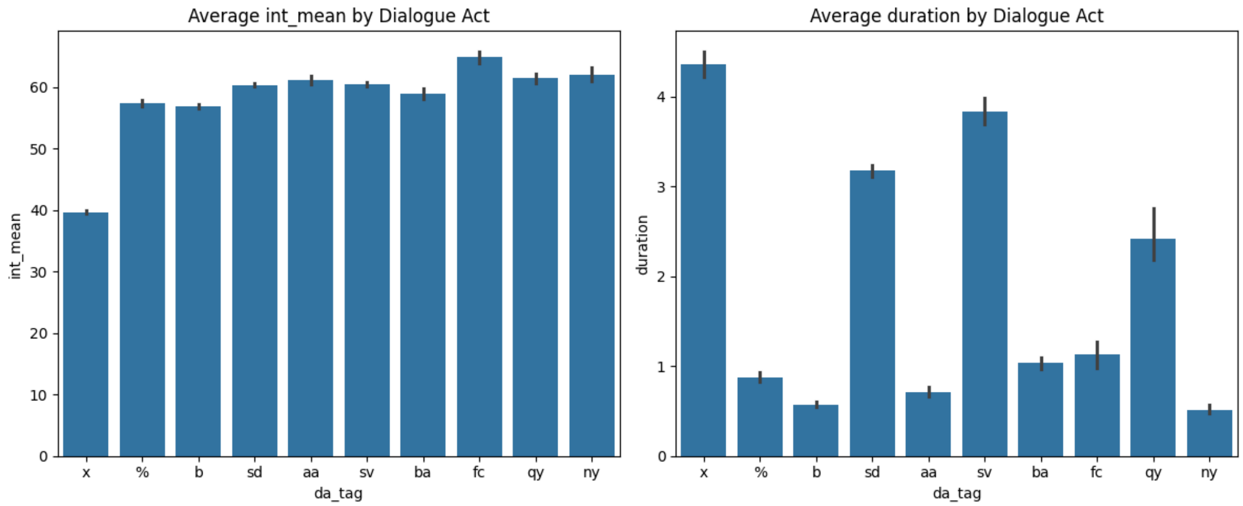
Conclusion:
The data does not support the hypothesis that assent is a distinctive marker of the aa dialogue act. Instead, the highest assent proportions occur in:ny (due to “yeah, no” constructions), and b (backchannels).
Therefore, LIWC “assent” is not a clean indicator of Accept/Agree (aa), but rather reflects conversational micro-responses and dispreferred answers.

Speech-based Feature Hypothesis:

The backchannel class b (e.g., “mm-hm”, “uh-huh”, “yeah”) will have lower average intensity (int_mean) and shorter duration (end_time − start_time) than other dialogue acts. Backchannels are brief listener signals produced with reduced effort, often low-amplitude and prosodically flat.

Code output:

```
===
Feature: int_mean
Mean (b):     56.8067
Mean (others):51.4039
Mann-Whitney U (b < others): U=23718576.0, p=1.000e+00
===
Feature: duration
Mean (b):     0.5717
Mean (others):3.4102
Mann-Whitney U (b < others): U=4411406.0, p=0.000e+00
```



From the left plot (Average int_mean by Dialogue Act) we can see that: Backchannels (b) have an average intensity of around 56–57 dB. Most other dialogue acts (sd, aa, sv, ba, fc, qy, ny) range from 58–65 dB. Especially, fc is the loudest (≈ 65 dB). (sd, aa) consistently have higher intensity than b. Only x is lower than backchannels.

Explanation:

Backchannels are meaningfully softer than typical dialogue acts. This strongly supports the hypothesis that people produce b acts quietly, reflecting their role as minimal, supportive listener feedback.

From the right plot (Average duration by Dialogue Act): Backchannels (b) have a duration around 0.6 seconds, one of the shortest durations in the corpus.
For Other categories:

* sv ≈ 3.8 s
* sd ≈ 3.2 s
* qy ≈ 2.4 s
* fc ≈ 1.1 s
* ba ≈ 1.1 s

Explanation:

Backchannels are much shorter than almost all dialogue acts. This matches real conversational behavior: backchannels are brief acknowledgments inserted between longer speaker turns. The only shorter one is ny, which also makes sense—short single-word responses like “yeah”. But even ny is a different functional category and still roughly comparable to b.

Conclusion:
Both predictions in your hypothesis are partially supported by the data:

* Backchannels are quieter than most dialogue acts.
* Backchannels are significantly shorter than other dialogue acts.

This reflects their discourse function: they are low-effort, low-prominence listener signals that acknowledge understanding without taking the floor.

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

b. Performance analysis.

b-1. Report the performance on validation set:

Model	Accuracy	F1
Speech	0.648361	0.221130
Text	0.854145	0.691088
Speech+Text	0.856703	0.691328

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

Based on the validation results:
The Speech + Text (gated fusion) model performs the best.
Although the improvement over the text-only model is modest, it achieves the highest validation accuracy and macro-F1.

Reason:
1. Text features carry most of the discriminative information

Dialogue acts in this dataset are heavily lexical (e.g., “yeah”, “right”, “I think…”, “why…?”), so text features—especially TF-IDF signals—provide strong cues. This is why the text-only model already performs very well (F1 ≈ 0.69).

2. Speech features provide complementary prosodic cues

Some dialogue acts, especially: backchannels (“uh-huh”, “mm-hm”), agree/accept, and yes-answers have identifiable acoustic patterns (low intensity, short duration, stable pitch…). The gated fusion model can exploit this extra signal only when helpful. This leads to a small but consistent improvement over text alone.

3. Gated fusion enables adaptive weighting

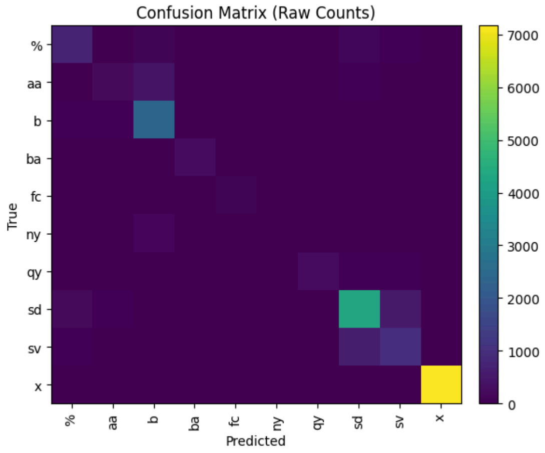
Unlike simple concatenation, the model uses a learnable gate:

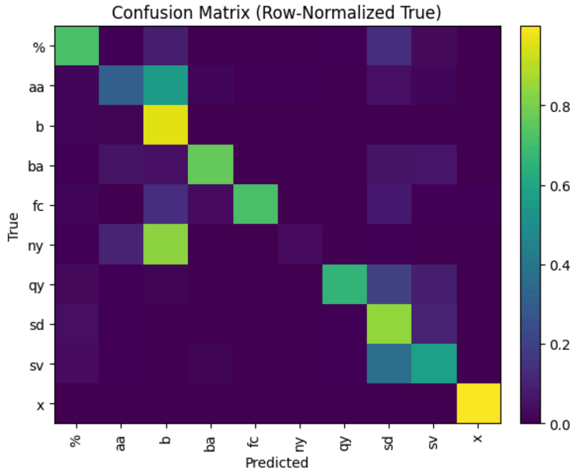
* If lexical content is very informative: rely on text
* If prosody gives clearer clues: rely on speech
* If both matter: combine them

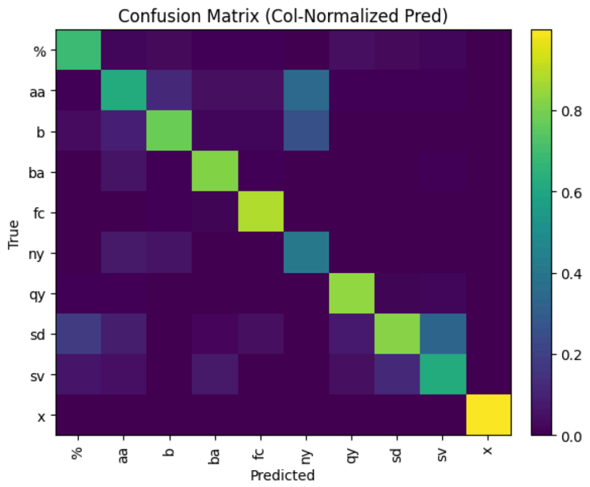
This dynamic fusion is why speech+text slightly surpasses the text-only model.

To sum up, The speech+text fusion model performs best because text features capture most of the lexical cues needed for dialogue act recognition, while speech features add complementary prosodic information (especially for backchannel-like acts), and the gated fusion mechanism allows the model to selectively combine both modalities.

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?

The easiest dialogue act classes to predict were “x” (non-verbal / silence) and “sd” (statement-declarative).
In both the raw and normalized confusion matrices, these classes show strong diagonal dominance, meaning that most of their samples were correctly classified.

*   “x” is easy because it has very distinct acoustic and textual patterns (long pauses, no lexical content, or silence markers). Both speech and text features can clearly separate it from spoken utterances.
*   “sd” is easy because it is the most frequent and lexically regular class. Declarative sentences often contain complete propositions (e.g., “I think that’s fine.”, “It looks good.”), which are easily captured by TF-IDF features and supported by steady prosody in speech.

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

The most difficult dialogue act classes to predict were “sv” (statement-opinion), “aa” (agree/accept), and “qy” (yes–no question).
These classes show large off-diagonal areas in the normalized confusion matrices, indicating frequent misclassification.
*   “sv” is often confused with “sd” (statement-declarative) because both are complete sentences with similar syntax and prosody. The only difference lies in subtle lexical cues like opinion markers (“I think”, “maybe”), which are harder to detect, especially when prosodic cues are weak.
*   “aa” overlaps with “b” (backchannel) and “ny” (yes-answer). These short affirmative responses (“yeah”, “right”, “yes”) have almost identical lexical forms and similar prosodic contours, making them challenging even for humans to distinguish.
*   “qy” is also difficult because declarative questions (“You’re coming?”) share lexical patterns with statements. Without strong rising intonation or punctuation cues, models often confuse them with “sd” or “sv”.

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

The most easily confused dialogue act pairs are (sv, sd) and (aa, b, ny), as seen from the strong off-diagonal regions in the confusion matrices.

* sv and sd (statement-opinion vs statement-declarative)
These two are the most frequently mixed-up classes. Both are full sentences and share nearly identical lexical and prosodic forms. The main difference lies in subtle opinion or subjectivity markers (e.g., “I think”, “maybe”), which are often absent or context-dependent.

The classifier confuses them because TF-IDF and acoustic features capture surface form, not pragmatic intent.

* aa and b and ny (agree-accept, backchannel, yes-answer)
These short utterances (“yeah”, “uh-huh”, “yes”, “right”) overlap heavily in both lexical and acoustic space. Their duration, pitch contour, and words are almost identical, so the model struggles to separate them even with prosody included.

The confusion reflects semantic and acoustic ambiguity, as well as limited context (single-word turns).

* fc and ba (conventional closing vs appreciation)
These are low-frequency classes that the model tends to misclassify into more common polite acts.

The errors likely come from class imbalance and data sparsity — the model doesn’t see enough examples to learn reliable boundaries.

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

Based on the error analysis, there are several promising directions to further improve the classifier:

1. Add contextual modeling

Many errors (e.g., aa vs b, sv vs sd) arise because the model sees each utterance in isolation. Incorporating previous and next turns (e.g., via RNNs or Transformer encoders over dialogue context) could help capture pragmatic cues and turn-taking signals.

2. Address class imbalance
Apply class-weighted loss functions or focal loss to ensure rare acts (e.g., fc, ba) are not overshadowed by frequent ones (sd, sv).
Oversampling techniques or data augmentation could also balance the training set.

3. Hierarchical or two-stage classification
Train a coarse-to-fine model:

* Stage 1: distinguish broad dialogue act families (statement / question / feedback).
* Stage 2: refine within each family (e.g., aa vs b vs ny).

This reduces confusion among semantically close classes.

4. Gated or attention-based fusion (beyond simple concatenation)
Extend the fusion model to allow cross-attention between speech and text embeddings, so the model can dynamically focus on the most informative modality per utterance.