COMS 6998: Advanced Topics in Spoken Language Processing Fall 2022
Homework 2 - Dialog Systems and Acts

Kyuran Kang – kk3583

Task 1: Dialogue Act Recognition

1. Feature Extraction

   a. Speech-Based:
   Praat/Parselmouth was used to extract speech-based features.

   b. Text-Based:
   The given LIWC features were used for text-based features.

2. Feature Description and Analysis

   a. Describe your custom feature sets, the reasoning behind choosing them and the techniques used to extract them. Also, include a description of any preprocessing you did.

      0) Preprocessing Prior to Feature Extraction

      Before extracting features, for an efficient feature extraction process, preprocessing was applied. First, I removed train rows with non-existing wav file dialog act ids. Then, dropped rows with classes that are not in the top 10 dialog act classes.

      1) Speech Features

      I extracted the following speech features:
      - Pitch (Min, Max, Mean, Sd)
      - Intensity (Min, Max, Mean, Sd)
      - Speaking Rate
      - Jitter
      - Shimmer
      - HNR

      The reasoning behind choosing these features was based on the top 10 dialog acts' properties and the experience from HW1, where these features were useful in detecting emotions. If we investigate the top 10 acts, we can assume that acts related to statements would have a relatively neutral pitch or intensity, and acts related to positive answers or questions might have a higher pitch or intensity. Speaking rates could also be different among the top 10 acts, for instance, people might speak questions faster than answering to questions. Jitter, shimmer, and HNR would be useful in differentiating stable from non-stable speeches. For instance, non-verbal speech acts, which include laughter and throat clearing, would be easily distinguishable from other acts by jitter, shimmer, or HNR.

Statement-non-opinion	sd	*Me, I'm in the legal department.*
Acknowledge (Backchannel)	b	*Uh-huh.*
Statement-opinion	sv	*I think it's great*
Agree/Accept	aa	*That's exactly it.*
Abandoned or Turn-Exit	% -	*So, -*
Appreciation	ba	*I can imagine.*
Yes-No-Question	qy	*Do you have to have any special training?*
Non-verbal	x	*[Laughter], [Throat_clearing]*

Yes answers	ny	*Yes.*
Conventional-closing	fc	*Well, it's been nice talking to you.*

After extracting speech features, I checked for null values in the datasets. For train set, I filled null values with average values of each feature of the same class. For test set, I dropped rows with null values. Then I scaled the datasets by Min-Max normalization. Lastly, I checked for features with zero variance and as there were no such features, I decided to use all extracted features.

2) Text Features

For text-based features, I used the given LIWC dataset, as LIWC features can illustrate people's attentional focus and these features can be useful in classifying the dialogue acts. People making statements would use more self-referencing words while in questions, words referencing others would be prevalently used. The verb tense can also be useful. For instance, in distinguishing statements with and without opinion, opinionated statements would be more likely to have future or present tense verbs, while statements explaining facts would generally have more past tense verbs.

Furthermore, LIWC features can be used to express emotions. Features like 'sad' and 'anger' would be indicators for speech acts related to expressing negative feelings. These features would appear more frequently in opinions than in positive speech acts such as Agree/Accept, Appreciation, and Conventional-closing.

Moreover, LIWC features can be used to classify speech acts with different thinking processes. The feature, 'cause' which indicates causal words like 'because', and the feature 'insight' for insight words like 'think' and 'know' may tell us about appreciation act or opinion act [1].

In addition, I added 'num_words' as a text-based feature, as some acts relating to statements would have more words compared to short answer acts like ny, b, aa, or %.

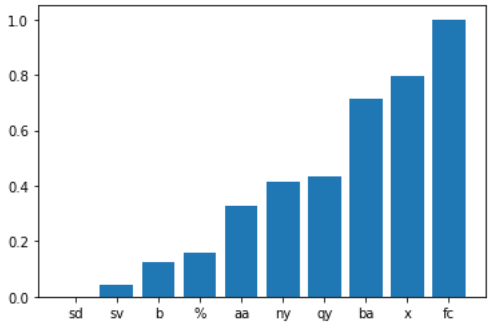
After extracting text features, I checked for null values and there were no null values in the text feature dataset. Then I scaled the data by Min-Max scaler. Also, I checked for features with zero variance, but there was no such feature, thus I decided to use all LIWC features.

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

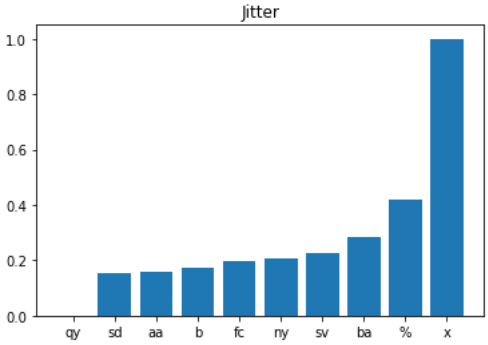
1) Speech Features

Hypothesis: *"Mean pitch would be useful in predicting the dialog act compared to Jitter or HNR"*

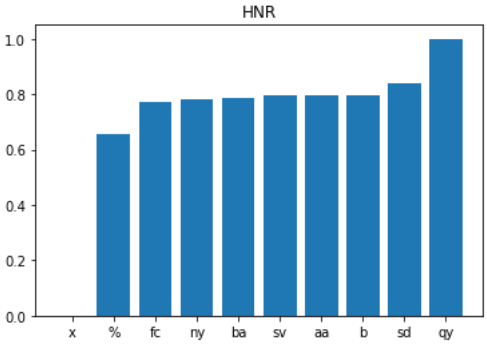
One can hypothesize that among the top 10 dialog acts, there could be noticeable differences or an order in average pitch among different acts. For instance, generally, sd (Statement non-opinion) and sv (Statement opinion) would have a lower pitch than question acts like qy. Then, statements referring to facts would have a lower pitch than opinion statements. Non-verbal speech would have a higher pitch than questions and so on. Moreover, considering the types of the top 10 acts, I think the jitters or harmonicities might be quite similar among the acts, except for the non-verbal act, x, which would have a higher average jitter and lower average harmonicity because it includes behaviors like throat clearing and laughter.



▲ Figure 1. Average Mean Pitch per Class



▲ Figure 2. Average Jitter per Class



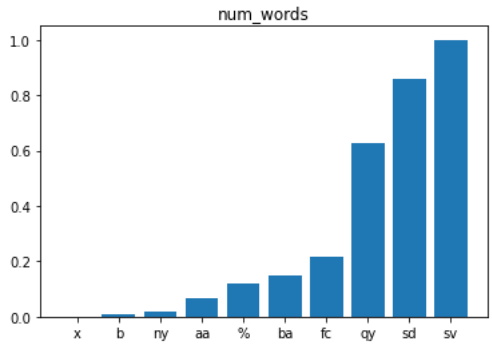
▲ Figure 3. Average HNR per class

I tested by calculating the average values for each feature per class and scaled the data to be between 0 to 1 to visually see the difference. In Figure 1. to Figure 3, one can observe that average per-class values of Mean Pitch are more spread out than those of the Jitter and HNR. While qy and x are easily distinguishable from other features by Jitter and HNR, other acts all have very similar average Jitters and HNRs, making them less useful in classifying other classes. Here, I calculated the average difference among the ten classes excluding the two largest differences; average Mean Pitches showed a 0.074 average difference while average Jitter and HNR showed 0.038 and 0.026, respectively. Thus, the test results are in accordance with the hypothesis. The result implies that the features related to stability of a speech are less critical in distinguishing speeches from each other except for the class x and qy, while pitch can be a relatively useful standard to classify different speech acts. This may be because Non-verbal acts are more likely to be prominently less stable and high in jitter and low in HNR compared to other classes. Yes-No-Question acts would generally be clearer and more stable, as questioning tends to be more confirmed and decided while answering acts or statement acts can be less stable than questioning but similarly moderate in stability.

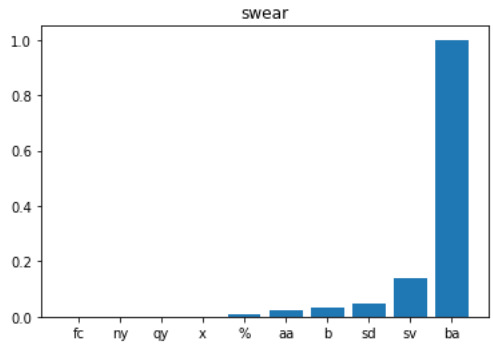
2) Text Features

Hypothesis: "*Number of words would be useful in predicting the dialogue act*"

The logic behind the hypothesis is that if we look at the types of top 10 acts, we can imagine each act having more words than the other ones. Thus, there can be noticeable differences in average word counts among different acts. For instance, statement acts would be longer than ba or ny, and ba would be longer than ny, and so on. Also, I think the 'swear' feature might not exist in most acts, making it a less useful component.



▲ Figure 4. Average Number of Words per Class



▲ Figure 5. Average 'swear' feature per class

In Figure 4 and 5, the average number of words are more spread out among the classes while average swear feature values are zero for fc, ny, qy, and x and the differences between %, aa, b, and sd are also very minor. The average difference among the ten classes excluding the two largest differences was 0.05 for num_words and 0.006 for the swear feature. Thus, the test results are in accordance with the hypothesis. The result may be due to people using more words to make statements than other classes and more words for statements with opinions; possibly because one may need to add a reason for their opinion. Also, it can be implied that questions are simpler than statements and that other speech acts (ba, %, aa, ny, and b) are usually short responses as they are mostly reactions to some statements or questions. b and ny, however, can be hard to be distinguished from each other, so we would need help from other features. On the other hand, swear words only appeared in certain acts; mostly in ba (Appreciation) and some in sv (Statement opinion). This would be because people usually don't use swear words in their conversations, and when they are used, they appear in only a few specific acts and situations.

3. Classification and Error Analysis

Using the feature sets, train machine learning classifiers to identify/predict the 10 most frequent dialog acts.

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

Trained Random Forest Classifiers for all feature types.

b. Describe the classification model(s) that you used and report the results for each feature set in tabular form.

Model	Accuracy	F1 (weighted)
Speech	0.69	0.62
Text	0.81	0.79
Speech + Text	0.80	0.78

I used Random Forest Classifier models. Random Forest Classifier is a tree-based ensemble model, and it is widely used for tabular data for its powerful performance. The model builds decision trees on different samples and takes their majority vote for classification [2]. I used scikit-learn to train and test the models. The model parameters used are shown below (referenced scikit-learn official document [3]):

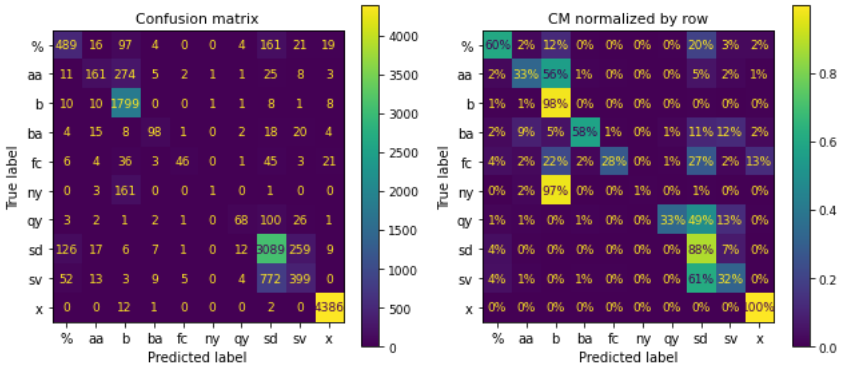
- min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='sqrt', max_leaf_nodes=None, min_impurity_decrease=0.0, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None

c. For the best-performing model:

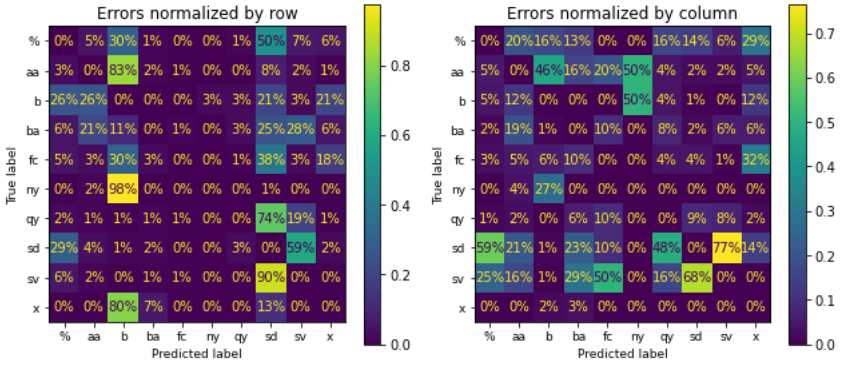
a. Which feature set performed the best?

Text feature set performed the best, while the performance of text model and text+speech model were very similar.

b. Show the confusion matrix.



Additionally, I made an error matrix by true label and an error matrix by the predicted labels.

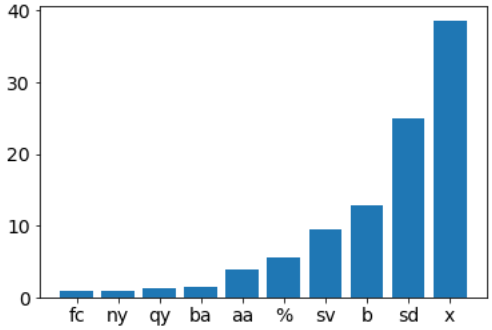


c. Which class(es) were easiest to predict? Why do you think they were easy?

The dialogue act 'x' was the easiest to predict (x is for Non-verbal class); approximately 100% of the x data were correctly classified. This would be because, for every scaled text feature, the Non-verbal class has the minimum value, which is zero, meaning the class Non-verbal has no text features; obviously a Non-verbal speech act won't have words in it. Thus, it would the easiest for the model to classify that out. Moreover, 39% of the train set was class x.

d. Which were the most difficult? Why do you think they were difficult?

The hardest one was 'ny' which is for the class Yes-answers; F1 score was the lowest. The model predicted ny to be b (Acknowledge (Backchannel)). However, b was well classified with the correctly classified rate of 98% while 97% of ny was predicted to be b. The reason would be that Acknowledge and Yes-answer classes have similar qualities in the text which would lead to similar text feature values and since b has 10028 rows while ny has only 729 rows of data in train set, the model would be trained to predict data as b when with similar text features were given. This is because tree-based models do not work well in imbalanced datasets. When a node is dominated by data from one class, the impurity criterion used to select a split point will determine the split as a good separation, when in fact, the examples from the minority class are being ignored [4].



▲ Train set data ratio by class

e. What were some common errors (e.g. confused classes)? Why do you think your classifier made these errors?

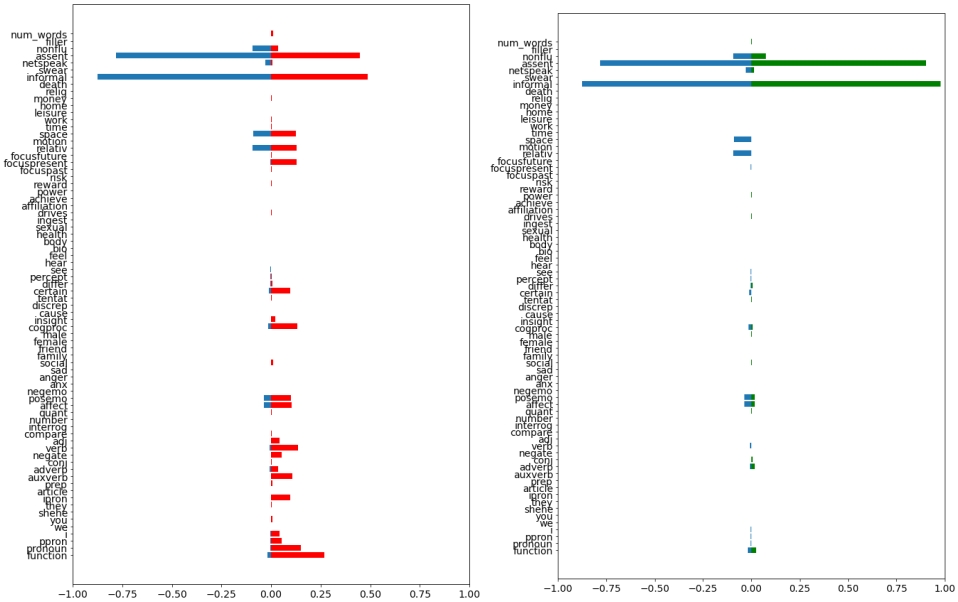
Besides the 'ny' class, the model misclassified 56% of 'aa' (Agree, Accept) class into class 'b'. 27% of 'fc' (Conventional-closing) was misclassified into 'sd' and 22% of 'fc' into 'b'. 49% of 'qy' (Yes-No-Question) was misclassified as 'sd', and 61% of sv (Statement-opinion) as sd.

The reason behind these errors would be that confused classes have similar textual traits. 'aa' and 'b' would have similar features as they are both short positive, or agreeing responses. Moreover, 'fc', 'qy', and 'sv' would have similar textual traits as 'sd' (Statement-Non opinion). For example, 'sd' and 'sv' are both statements so they would have commonalities and we can imagine 'fc' (Conventional-closing) having traits of statements. However, 'b' (98% of predictions were correct) and 'sd' (88% correct) was well predicted. This would be because tree models are more likely to predict similar data into a more dominant class.

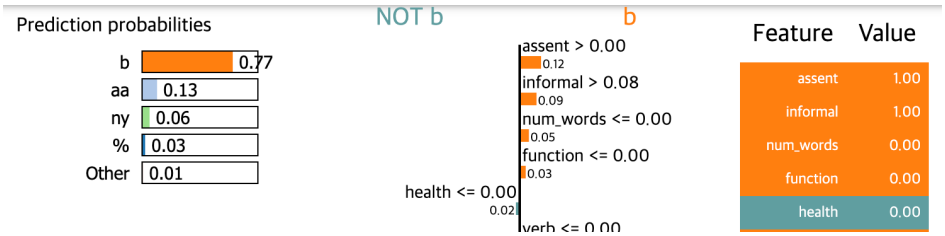
[Optional Part]

I investigated some wrong samples and compared average feature values.

1) 'ny' and 'aa' misclassified as 'b'

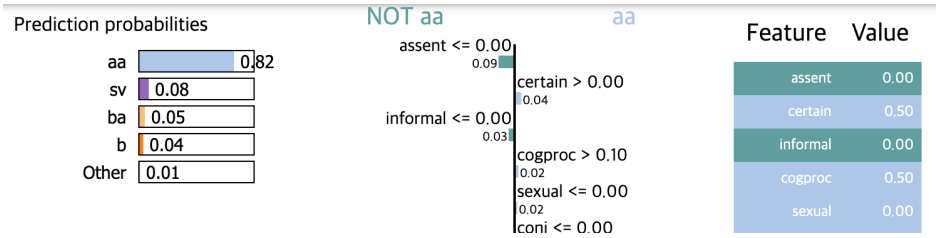


The above graph on the left shows the average of each text feature value for class ‘aa’ (red) and class ‘b’ (blue). ‘b’ has no to little average values for most text features while having high values for feature assent and informal, on the other hand, ‘aa’ has values in some other features such as function, pronoun, and cogproc. The graph on the right shows class ‘ny’ (green) and class ‘b’ (blue). ‘ny’ also has very high value for assent and informal just like class ‘b’.



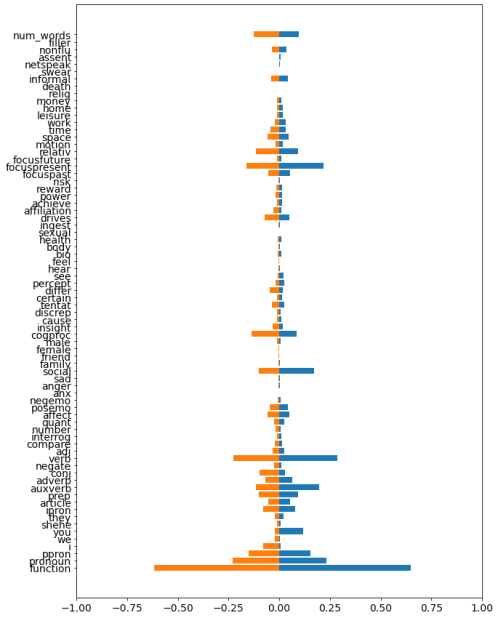
Above is a sample of misclassified ‘aa’ data. It is observed that an ‘aa’ sample deviating from average feature values and only having high assent and high informal feature values was classified as ‘b’.

Below is a sample of correctly classified ‘aa’ data. This sample was correctly classified having higher value in other features, certain and cogproc.



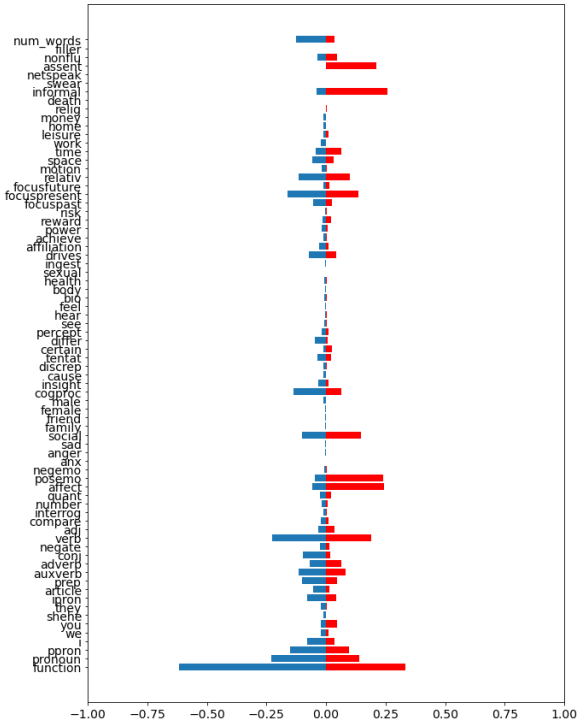
This can imply that a sample with high assent and high informal feature values would be classified into class ‘b’ which is more dominant.

2) ‘qy’ misclassified as ‘sd’

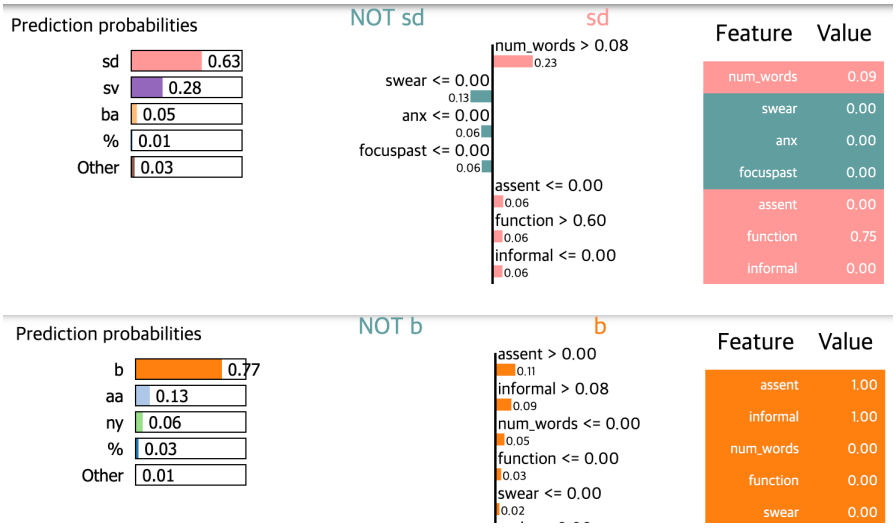


The above graph represents the average of each text feature value for class ‘qy’ and class ‘sd’. Both classes show very similar feature patterns, which would lead to misclassification. From this, we can say that the questions and statements with no opinion are very similar in language and since there are way more ‘sd’ class samples (25% of total data) in train set (‘qy’ class: 1.3% of total data), the model is more likely to predict a sample into ‘sd’ class.

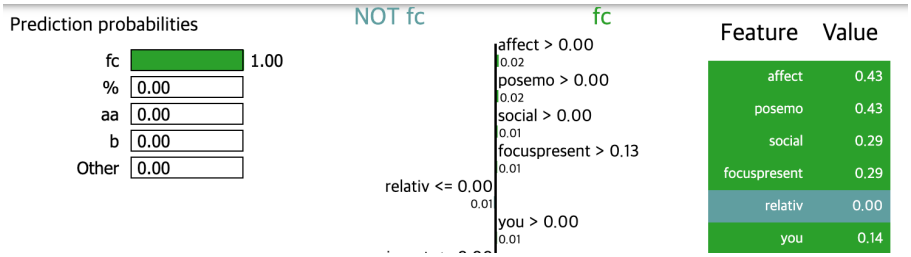
3) ‘fc’ misclassified as ‘sd’, and ‘b’



Above is a graph for class ‘fc’ (red) and ‘sd’ (blue). In comparison to ‘sd’, ‘fc’ has higher assent and swear average values and is low on num_words and function features. In the below sample, we can see that ‘fc’ is misclassified as ‘sd’ when having a high num_words value.



Above is an example of ‘fc’ misclassified as ‘b’. A sample with high assent and informal features values was classified as ‘b’. This behavior is analogous to ‘ny’ and ‘aa’ being misclassified as ‘b’. But ‘fc’ having other features was well classified as shown in the below sample.



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

First, we can try to obtain more samples to build a more general model and resolve the imbalanced data issue. Second, as obtaining more data often can be a time-consuming and inexpensive process, we can try to randomly oversample minor classes or oversample and under-sample at the same time or use advanced methods such as SMOTE to oversample the data. Another option might be to use different feature sets or add features that can help distinguish classes that were hard to discriminate from each other. For instance, adding features related to punctuations can help distinguish questions and statements. Additionally, selecting more useful speech features can help to distinguish the data. Another option can be to use different types of models and compare them to get the best model for this dataset.

Bibliography

[1] N. Novielli and C. Strapparava, “The role of affect analysis in dialogue act identification,” IEEE Transactions on Affective Computing, vol. 4, no. 4, pp. 439–451, 2013.

[2] S. E. R, “Random Forest: Introduction to random forest algorithm,” Analytics Vidhya, 21-Jun-2022. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/. [Accessed: 20-Nov-2022].

[3] “Sklearn.ensemble.randomforestclassifier,” scikit. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html?highlight=randomforest#sklearn.ensemble.RandomForestClassifier. [Accessed: 20-Nov-2022].

[4] J. Brownlee, “Cost-sensitive decision trees for imbalanced classification,” MachineLearningMastery.com, 20-Aug-2020. [Online]. Available: https://machinelearningmastery.com/cost-sensitive-decision-trees-for-imbalanced-classification/. [Accessed: 20-Nov-2022].