Written Reporting

1. Summarize the task based on the paper in 1-2 paragraphs. What are some assumptions made? Can you think of other uses for this decision list approach?

Yarowsky's Decision lists for lexical ambiguity resolution: Application to Accent Restoration in Spanish and French explored accent disambiguation. However, the decision list approach as a whole aims to identify features that demonstrate higher or lower likelihoods for a binary outcome relative to the set. The process explored in this paper specifically deals with accent disambiguation and begins with identifying a series of binary accent ambiguity counts and generating a histogram. Following this, the paper honed in on a single binary accent ambiguity problem and collected a variety of labeled instances to be used as a training set. These labeled instances were then utilized to form collocation distributions, which entailed measuring the frequencies of features for both classes, such as: words +/- k, parts of speech, and bigrams. Some of these features are combined before taking the absolute log likelihood of each feature. These log values were then sorted by ranking them from highest to lowest. The features with logs that were the highest were then utilized to generate the decision list. Once generated, the test set can be run against it and for each instance, the likelihood of each class will be determined by the feature log values, and assigned classification.

One assumption made from this method is word dependence. The target word was not taken at face value as features included bigrams and +/- k and their parts of speech. It expresses that a word's classification is not independent, but rather dependent on the words surrounding it. Furthermore, this paper initially considered words found in the +/- k window and bigrams at various offsets, suggesting that they believed those features to be the most likely to be useful for classification. Overall it was interesting to note that this approach is very different from Naive Bayes, which assumes word independence. Another use for this decision list approach could be generating mailing lists, as certain behavioral features could be utilized to indicate whether participants are likely to interact with the mail received. This can in turn express whether an individual should be added to the mailing list, which can be vital if there are limited spots on the mailing list. In that specific case, decision lists could even help to prioritize which individuals should be on the mailing list over others based on their likelihood to interact with the mail content.

2. Explain what you did, conceptually, and any decisions you took.

For this problem, one of the first major aspects to tackle was preprocessing and organization of the data, as we were already provided the training and testing instances. To achieve this task, we wrote an algorithm that could take in bass.trn, sake.trn, bass.tst, or sake.tst and then dynamically generated a json file containing the instance set to lowercase and had punctuation removed, the label and associated word +/- k features for each instance. For this algorithm, we assumed that file passed had one instance per line and were formatted as "(*)[class]: [instance words with target inside]". We believed it would be easier to store the data in an organized manner with which we could later access once closing the input file. This then enabled our group to split the code into various smaller and cleaner methods.

Once processed, we selected our features. These features included all +/- k words surrounding the target of “bass” or “sake” and the parts of speech of those words. But in our approach we tried not to complicit it by selecting only +/-2 words, +/-1 words and parts of speech for -1 words. Then we generated counts for each of the features for each class to create the collocation distribution. Afterward, we calculated the log likelihoods for each and generated the decision list. From there this decision list was compared against the test set. For each of the instances, each of the features’ associated log values were selected and the max value was found. This max value was associated with one of the predicted classes. This predicted class was then compared to the actual. For a very few instances, there was no actual class provided and so we had to decide on what the default class would be. We compared the effect on our confusion matrix data and explored that below. Our program then outputs this confusion matrix.

3. Report on the top-10 decision rules each for both BASS vs. SAKE.

SAKE:
[('of', [8.233768709217097, 'sake']), ('of_the', [6.641182169740591, 'sake']), ('DET_the', [4.958261794364674, 'sake']), ('of_a', [4.948759890378168, 'sake']), ('own', [4.875197323201151, 'sake']), ('of_peace', [4.836281906951478, 'sake']), ('of_our', [4.795790545596741, 'sake']), ('ADJ_own', [4.795790545596741, 'sake']), ('of_their', [4.7535901911063645, 'sake']), ('its_own', [4.394449154672438, 'sake'])]

BASS:
[('player', [5.638354669333745, 'bass']), ('sea', [5.351858133476067, 'bass*']), ('NOUN_sea', [4.875197323201151, 'bass*']), ('largemouth', [4.836281906951478, 'bass*']), ('lines', [4.61512051684126, 'bass']), ('and_drums', [4.564348191467836, 'bass']), ('PROPN_sea', [4.394449154672438, 'bass*']), ('guitar', [4.330733340286331, 'bass']), ('PROPN_largemouth', [4.189654742026425, 'bass*']), ('played', [4.110873864173311, 'bass'])]

In the SAKE, as you can see that the topmost decision rules are all **“SAKE” (purpose**) sense. 8/10 words are bigram words are representing the top 10 words in SAKE sense. This denotes that +/- 2 words nearer to the SAKE can almost tell if the sentence belongs to the **“purpose”** sense or not. There are no food senses in the top 10 liklihoods list, that denotes that food sense needs even more features leading to trigrams,+/- 6 words and also more parts of speech tagging.

In the BASS, you can see that almost single words can determine the meaning of BASS sense. The following words (Guitar, played, player, lines) belong to **BASS** sense and they are unigram words, which denotes the music sense can be derived almost using the unigram words. When you are checking the other sense (sea, largemouth, NOUN_sea, PROPN_sea) are also almost unigram words but parts of speech attached to this can determine the meaning well for the **BASS*** (**fish**) sense.

4. Report on performance for each test set:

a. **For each case, compare against a baseline that assigns the most likely sense found in the training set, to all instances in the test set for a given word. For bass vs sake: What is the baseline sense label? What is the result of this simple baseline on the test set? Use a table.**

b. **What improvement, if any, did you get over this baseline? Report on classification accuracy.**

c. **Include confusion matrices for BASS and SAKE and your observations. Also include 3 example instances from each test set that were correctly vs. 3 incorrectly classified per case.**

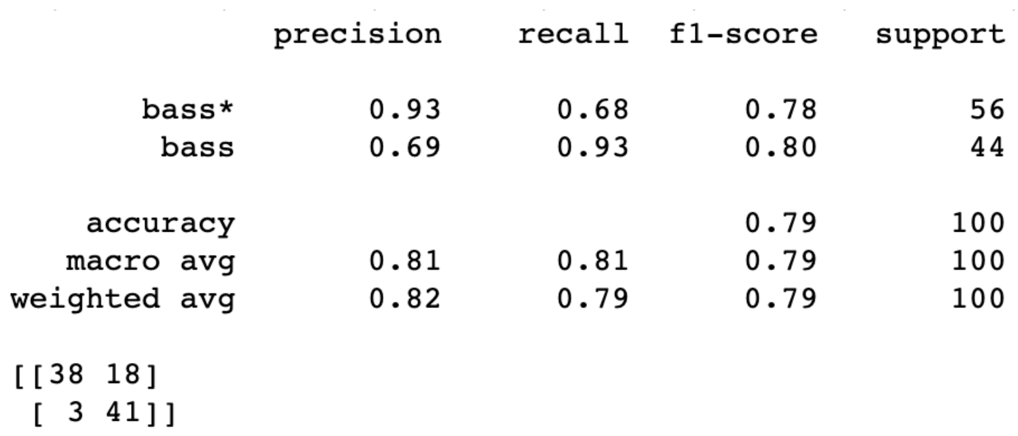
d. **Compute two of these metrics per case and include them in the table. Explain how you operationalized the chosen metrics.**
   i. **Compute % absolute change in accuracy compared to the baseline.**
   ii. **Compute % error reduction over baseline.**
   iii. **Macro-averaged precision (or recall) weighing classes equally**

5. Provide a written discussion where you reflect upon results and reason about them. For example: What influenced the results on performance metrics? What do the confusion matrices tell you? For examples provided, what went right/wrong? If you encountered any concerns along the way, mention those too

6. We will run your code for *bass* and *sake* so clearly describe to us how to update the train and test sets in your notebook.

- Train and Test data can passed on the main function to run the entire program.

- **bass* as bias**



```
              precision    recall  f1-score   support

       bass*       0.93      0.68      0.78        56
        bass       0.69      0.93      0.80        44

    accuracy                           0.79       100
   macro avg       0.81      0.81      0.79       100
weighted avg       0.82      0.79      0.79       100

[[38 18]
 [ 3 41]]
```

- **bass as bias**

```
              precision    recall  f1-score   support

       bass*       0.91      0.77      0.83        56
        bass       0.75      0.91      0.82        44

    accuracy                           0.83       100
   macro avg       0.83      0.84      0.83       100
weighted avg       0.84      0.83      0.83       100

[[43 13]
 [ 4 40]]
```

- **Sake* as bias**

```
              precision    recall  f1-score   support

       sake*       0.97      0.99      0.98        94
        sake       0.75      0.50      0.60         6

    accuracy                           0.96       100
   macro avg       0.86      0.74      0.79       100
weighted avg       0.96      0.96      0.96       100

[[93  1]
 [ 3  3]]
```

- **Sake as bias**

```
              precision    recall  f1-score   support

       sake*       0.95      1.00      0.97        94
        sake       1.00      0.17      0.29         6

    accuracy                           0.95       100
   macro avg       0.97      0.58      0.63       100
weighted avg       0.95      0.95      0.93       100

[[94  0]
 [ 5  1]]
```