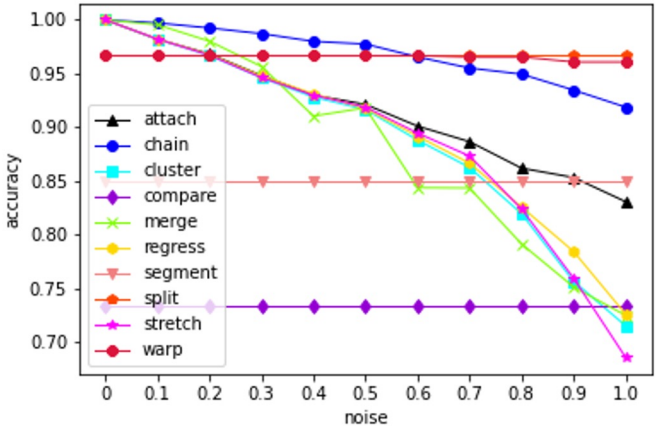
Drift Correction: Initial Results

Luhang Sun, Deka Popov, Bret Miller, Sidney Liu

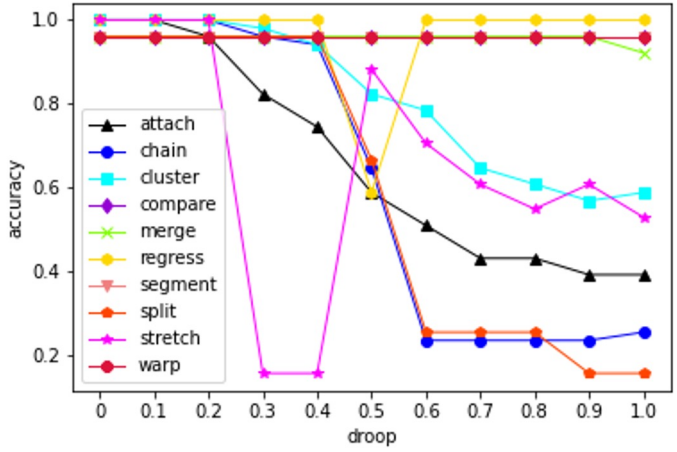
Synthetic Dataset

```
for i in error_type:
    for j in range(0, 11):
        for index in range(100):
          if error_type == 0: # noise
            synth_fixations = correction.generate_fixations_left_skip_regression(aois_with_tokens)
            error_trial = correction.error_noise(j/10, random.randint(0, 50), j/10, synth_fixations)

          elif error_type == 1: # shift
            synth_fixations = correction.generate_fixations_left_skip_regression(aois_with_tokens)
            line_ys = np.array(synth_fixations)[:, 1]
            error_trial = correction.error_shift(j/10, line_ys, synth_fixations)

          elif error_type == 2: # droop
            synth_fixations = correction.generate_fixations_left_skip_regression(aois_with_tokens)
            error_trial = correction.error_droop(j, synth_fixations)

          elif error_type == 3: # offset
            synth_fixations = correction.generate_fixations_left_skip_regression(aois_with_tokens)
            error_trial = correction.error_offset(j, j, synth_fixations)

          else: # no error
            synth_fixations = correction.generate_fixations_left_skip_regression(aois_with_tokens)
```

Algorithm Accuracy on Synthetic Data (noise, droop)

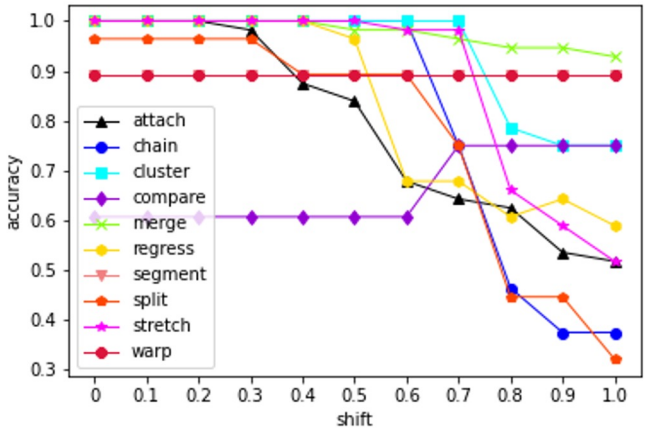



```
mean attach: 0.9167121212121212
mean chain: 0.9689090909090908
mean cluster: 0.8893181818181818
mean compare: 0.7333333333333333
mean merge: 0.8833030303030303
mean regress: 0.8946818181818181
mean segment: 0.85
mean split: 0.9666666666666667
mean stretch: 0.8892727272727272
mean warp: 0.9653939393939394
```

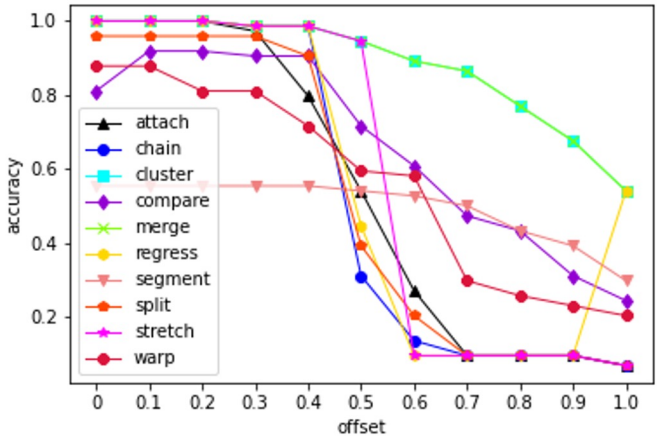



```
mean attach: 0.661319073083779
mean chain: 0.6131907308377896
mean cluster: 0.8128877005347593
mean compare: 0.9607843137254902
mean merge: 0.9572192513368984
mean regress: 0.9625668449197861
mean segment: 0.9607843137254902
mean split: 0.5953654188948306
mean stretch: 0.6541889483065954
mean warp: 0.9607843137254902
```

Algorithm Accuracy on Synthetic Data (shift, offset)

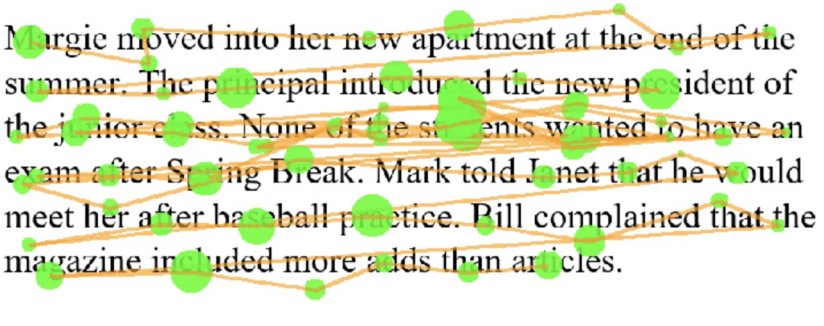


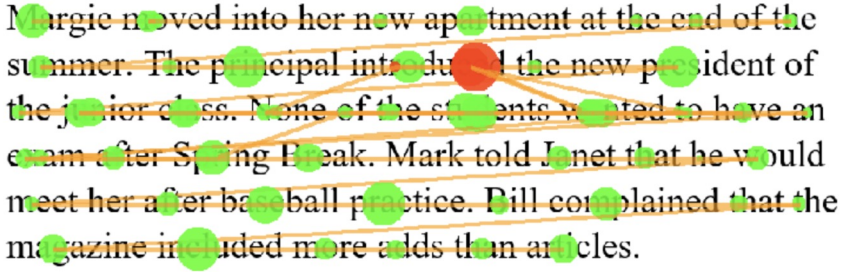

```
mean attach: 0.7905844155844156
mean chain: 0.814935064935065
mean cluster: 0.935064935064935
mean compare: 0.6590909090909091
mean merge: 0.9772727272727273
mean regress: 0.8327922077922078
mean segment: 0.8928571428571429
mean split: 0.7727272727272727
mean stretch: 0.8847402597402597
mean warp: 0.8928571428571429
```



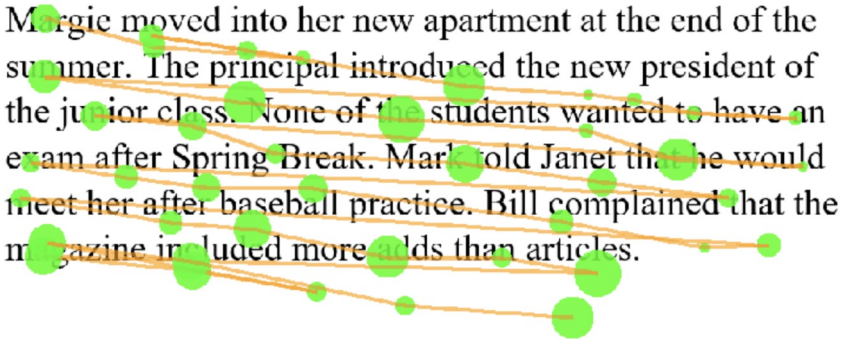

```
mean attach: 0.5393120393120393
mean chain: 0.5245700245700246
mean cluster: 0.8783783783783784
mean compare: 0.6584766584766585
mean merge: 0.8783783783783784
mean regress: 0.5761670761670762
mean segment: 0.4963144963144963
mean split: 0.5171990171990172
mean stretch: 0.5786240786240786
mean warp: 0.5687960687960688
```

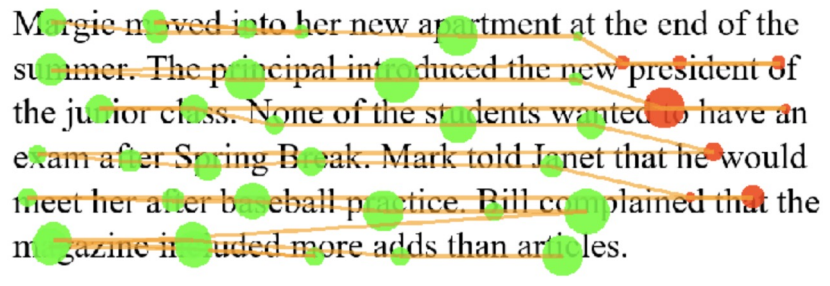
Synthetic Data Correction: Noise (chain algo)



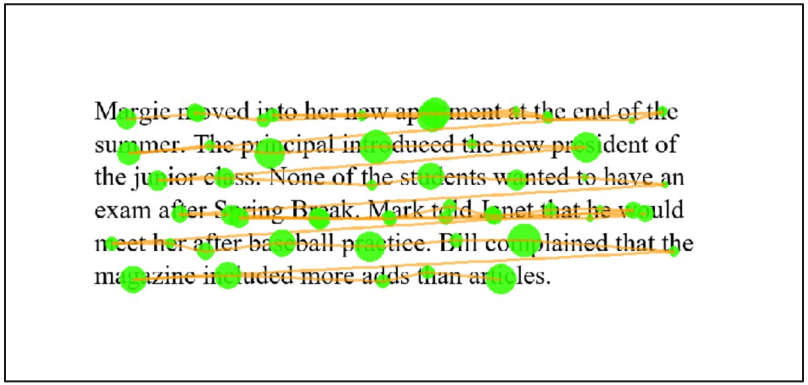


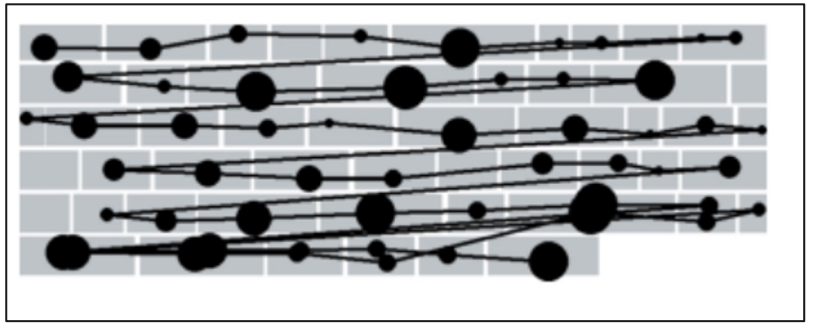
Synthetic Data Correction: Droop (regress algo)





Input Generalization for Neural Network



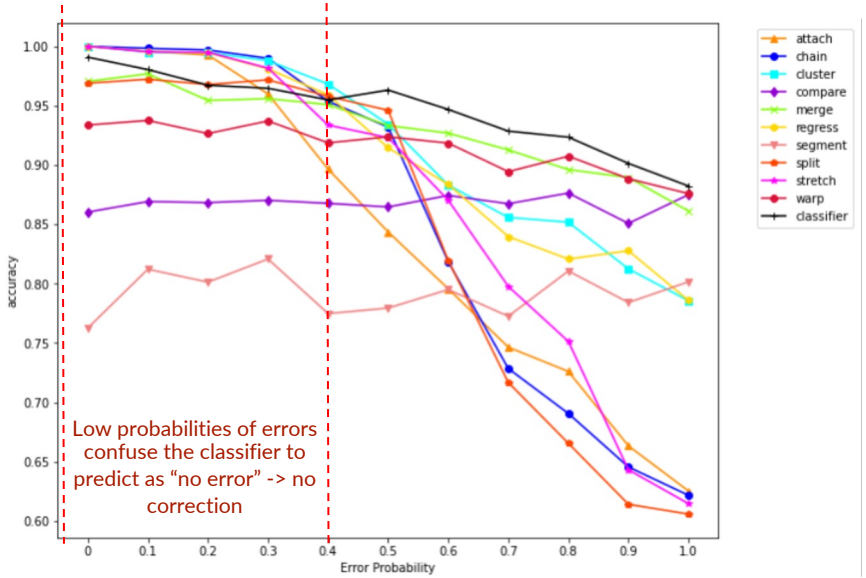


- “Grey-box” to represent text
- Trimmed image

NN Classifier Accuracy

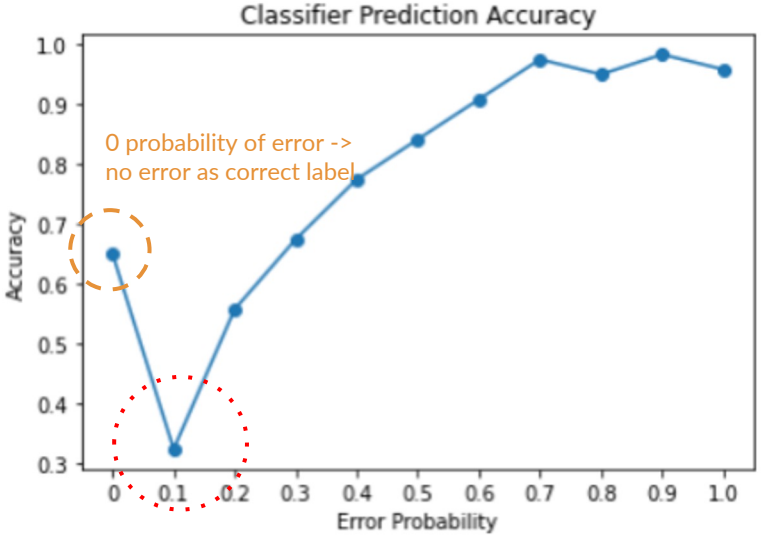
- Epochs = 30; batch_size = 30
- Loss in validation/testing accuracy after switching to the “grey-box” model
- ~97% training, ~75% testing accuracy

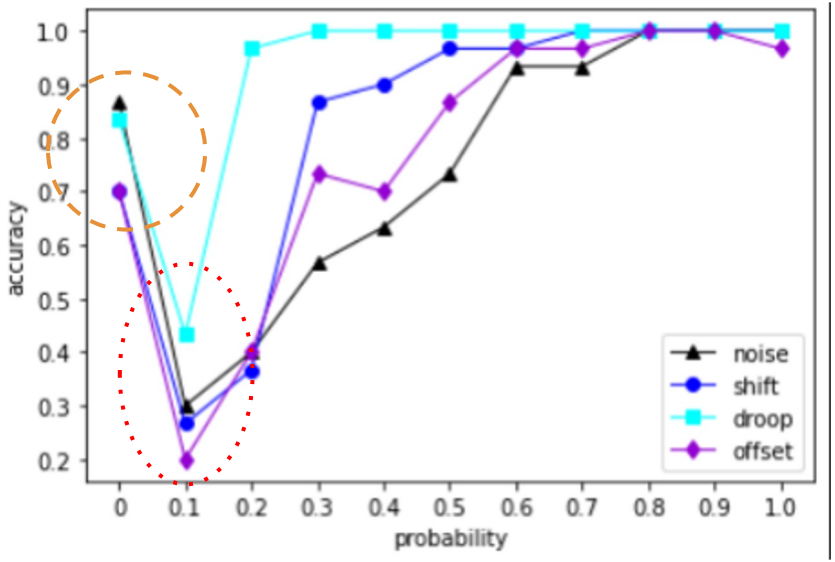
Did classification help correction?



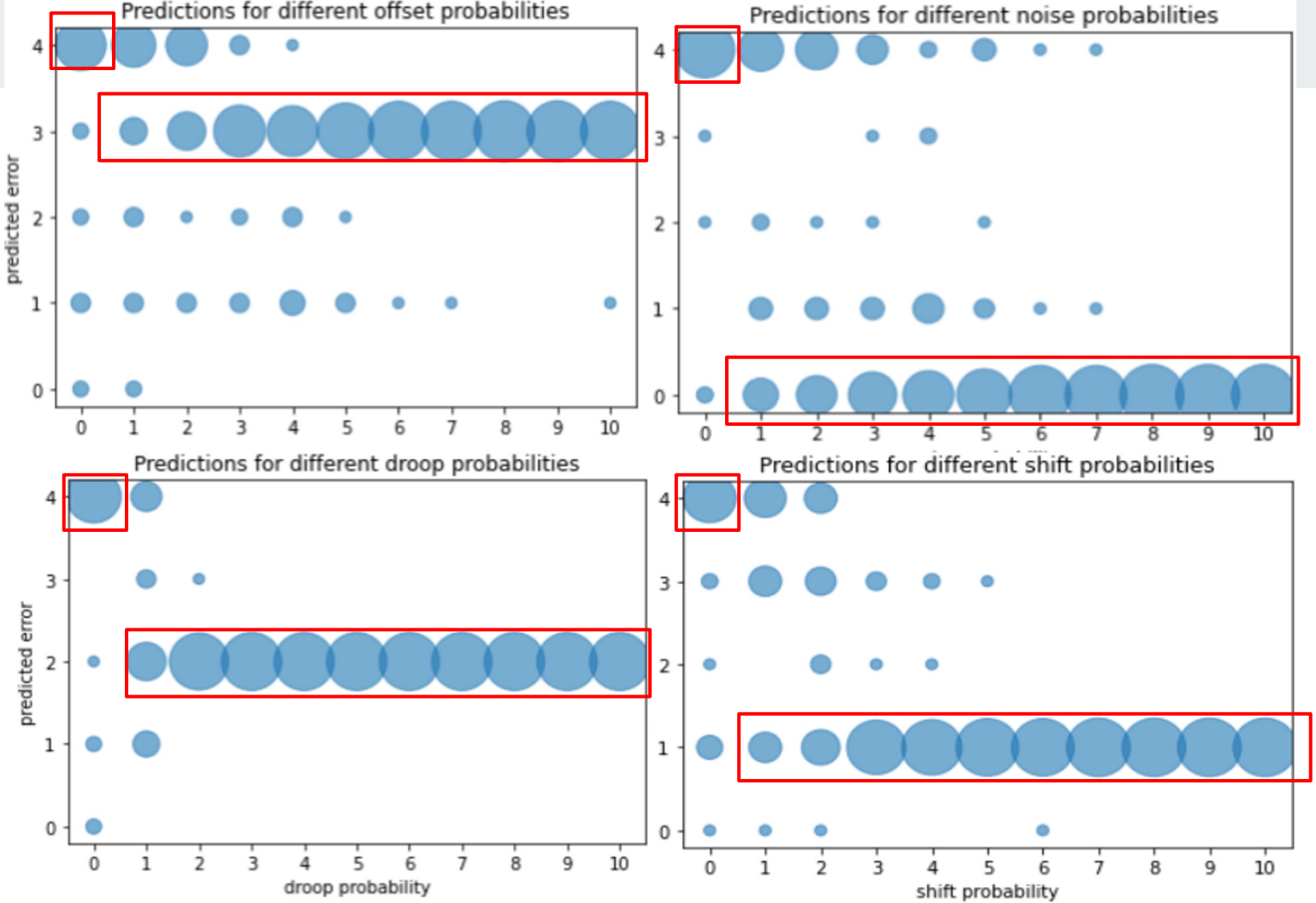
- The average of all four types of errors repeated for 30 times (120 samples) at each probability
- With random regression and skipping
- The relatively low accuracy of the model with classifier is mainly due to low prediction accuracy when probability is low

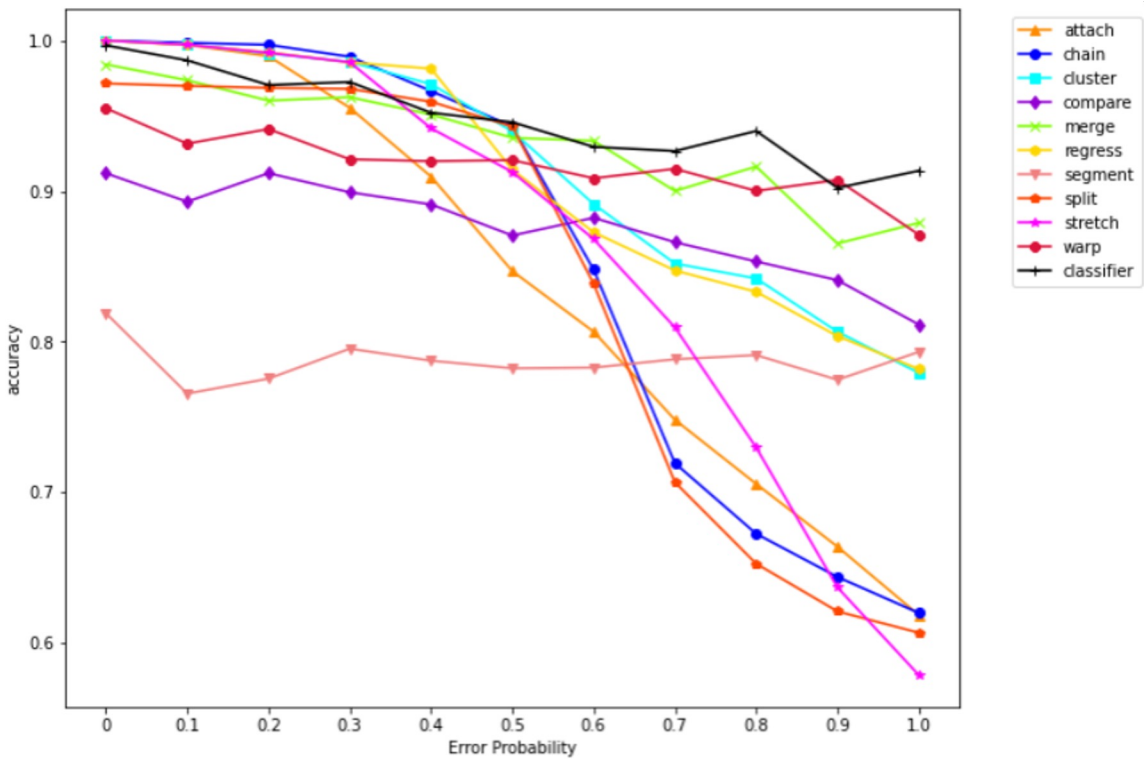
Classifier's confusion at low error probability





The NN classifier tends to predict trials with low error probability as "no-error"


Predictions for different offset probabilities
Predictions for different noise probabilities
Predictions for different droop probabilities
Predictions for different shift probabilities
predicted error
droop probability
shift probability



Use “chain” when predicted as “no error”

Golden Set Generation

- Used Fix8 to correct each trial
- Created a set of Json files to be used as the “ground truth” for our dataset
- Consisted of 15 different trials to comprise the Golden Set
- Was time intensive, but gave insight into how different peoples reading behavior is
    - Helped us understand the way that different real world drift phenomena occur in actual eye tracking trials

Correction Accuracy on GazeBase

- We utilized the original algorithms to see how they would perform on real world data
- Based on Carr's paper, we knew that the accuracy would not be as good as on the synthetic data
- At first, I was receiving 0,1,2% in terms of accuracy, but one of my partners figured out where the bug was
- In addition, I ran every single trial, through every single algorithm to find the performance across all the trials
    - This allowed for us to compare how they fared against each other when it came to working with real world data, which usually had every type of error present.

Correction Accuracy