CS323: INTRODUCTION TO NLP

Team Project

Paraphrase Detection Model and Label Noise Analysis

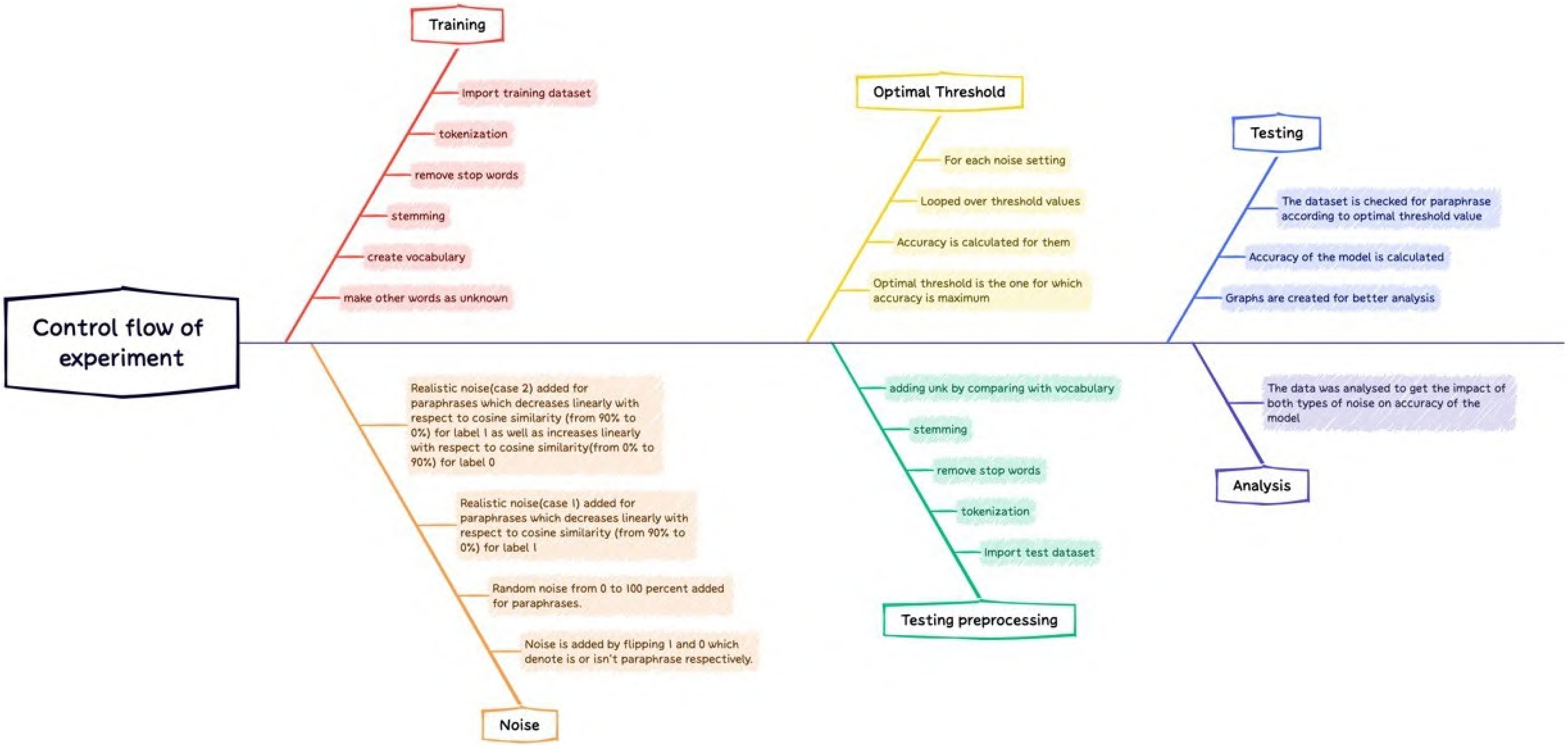
↘

Pitch

We analyze the impact of label noise on the performance of the paraphrase detection model. i.e. We will compare the sensitivity to label noise (robustness) in our paraphrase detection model

Colab Link: https://colab.research.google.com/drive/1zuXoMMi2WG1q_AgCDiJUIxCUlavKWZ4h?usp=sharing

Introduction ↘



Control flow of experiment

Training

- Import training dataset
- tokenization
- remove stop words
- stemming
- create vocabulary
- make other words as unknown

Noise

- Realistic noise(case 2) added for paraphrases which decreases linearly with respect to cosine similarity (from 90% to 0%) for label 1 as well as increases linearly with respect to cosine similarity(from 0% to 90%) for label 0
- Realistic noise(case 1) added for paraphrases which decreases linearly with respect to cosine similarity (from 90% to 0%) for label 1
- Random noise from 0 to 100 percent added for paraphrases.
- Noise is added by flipping 1 and 0 which denote is or isn't paraphrase respectively.

Optimal Threshold

- For each noise setting
- Looped over threshold values
- Accuracy is calculated for them
- Optimal threshold is the one for which accuracy is maximum

Testing preprocessing

- adding unk by comparing with vocabulary
- stemming
- remove stop words
- tokenization
- Import test dataset

Testing

- The dataset is checked for paraphrase according to optimal threshold value
- Accuracy of the model is calculated
- Graphs are created for better analysis

Analysis

- The data was analysed to get the impact of both types of noise on accuracy of the model

Model Setup

Step #1: **Preprocessing of the train and test set**	• Removes the stop words • Performs lemmatization • Classify rare words into UNK • Induce Noise • Vectorization
Step#2: **Calculate threshold and accuracy**	• Loop over different threshold values • Compute prediction score using train dataset • For the highest prediction score, select threshold as optimal.
Step#3: **Output for test dataset**	• Compute accuracy for test set for further analysis, using the optimum threshold found above. The f1-score is also computed for the test dataset as an extra metric for analysis.

Dataset

MRPC
(Microsoft Research Paraphrase Corpus)

What is MRPC

MRPC is a corpus consists of 5,801 sentence pairs collected from newswire articles. Each pair is labelled if it is a paraphrase or not by human annotators.

Train and Test Division

The whole set is divided into a training subset (4,076 sentence pairs of which 2,753 are paraphrases) and a test subset (1,725 pairs of which 1,147 are paraphrases).

Format of the dataset

Quality ID1 ID2 String1 String2
Quality is 1(paraphrase) or 0(not paraphrase)
ID's are irrelevant to the experiment.
Strings are the two sentences.

Noise Implementation

RANDOM NOISE

- Performed for various noise percentages.
- Randomly chooses some indexes which had label 1 (were paraphrases) , inverts the corresponding labels to 0.

REALISTIC NOISE (CASE 1)

- Percent noise added depends on the cosine similarity. Different percentage for different similarity ranges.
- Noise added for paraphrases decrease linearly with respect to cosine similarity (from 90% to 0%) for label 1

REALISTIC NOISE (CASE 2)

- Noise added for paraphrases decrease linearly with respect to cosine similarity (from 90% to 0%) for label 1 as well as increase linearly with respect to cosine similarity(from 0% to 90%) for label 0

Noise Analysis

↘

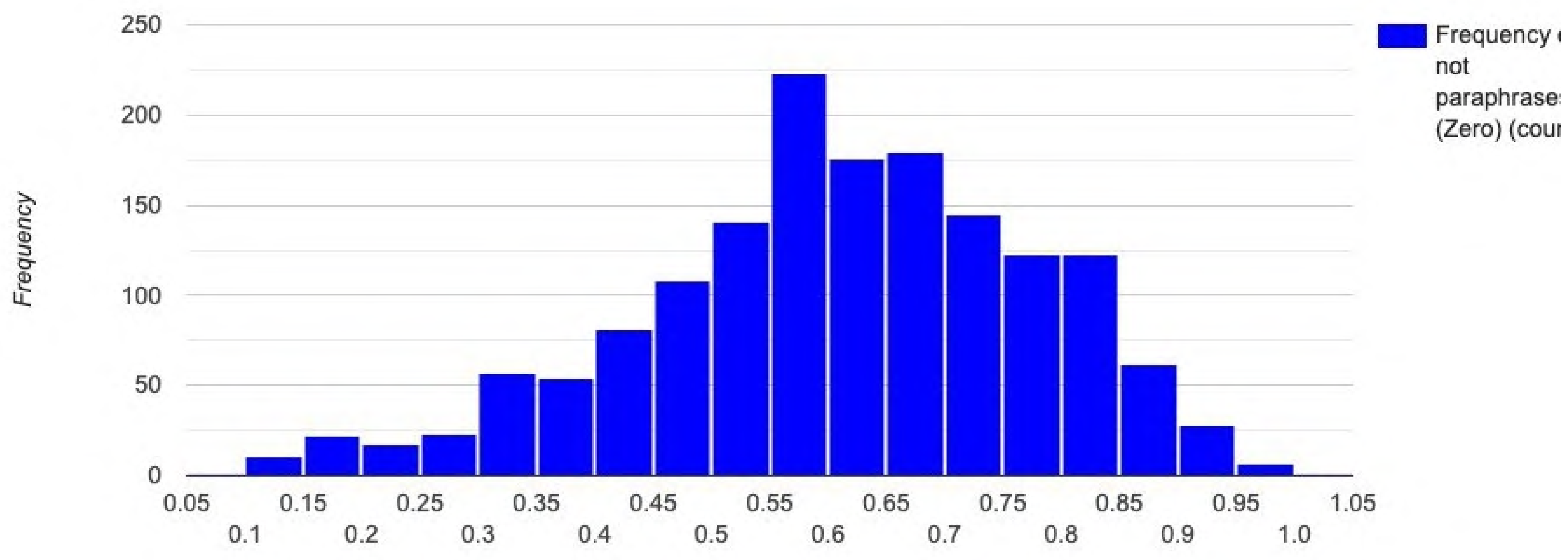
Random Noise

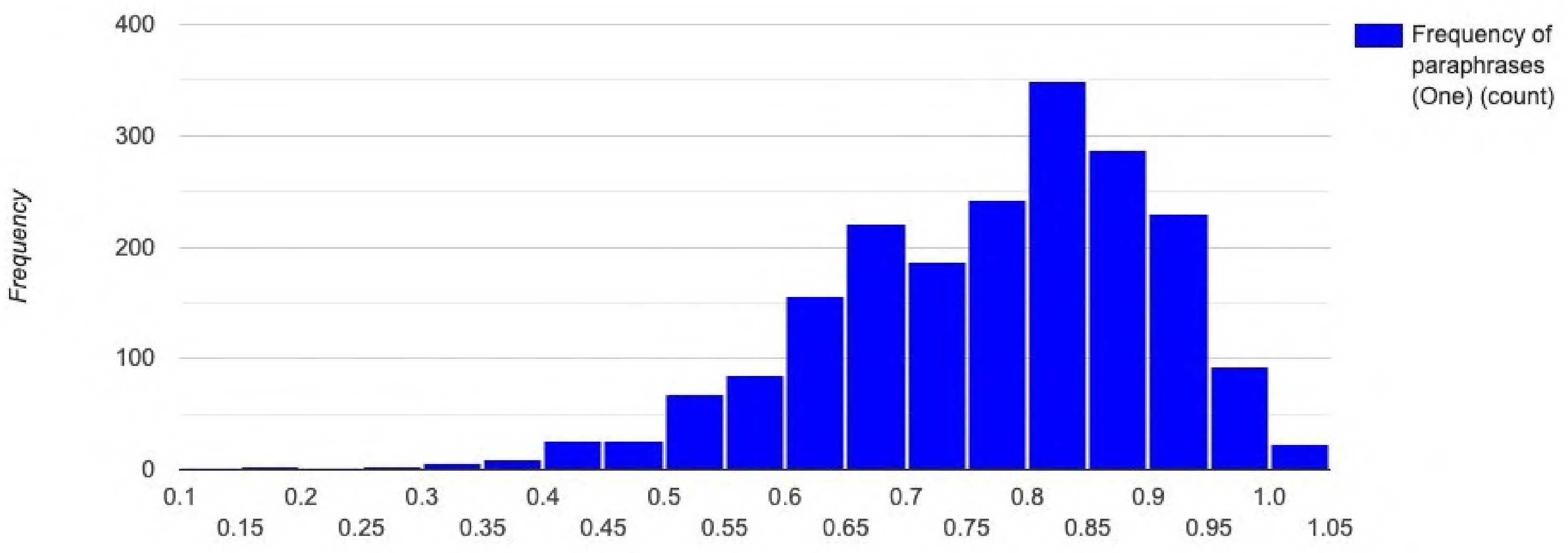
Noise Percent	Optimum Threshold	Accuracy	F1-score
0	0.605	70.84%	0.786
10	0.575	71.53%	0.798
20	0.515	72.23%	0.815
30	0.605	70.84%	0.786
75	0.5	72.05%	0.816
80	0.855	47.88%	0.372
90	0.995	33.91%	0.012

Realistic Noise

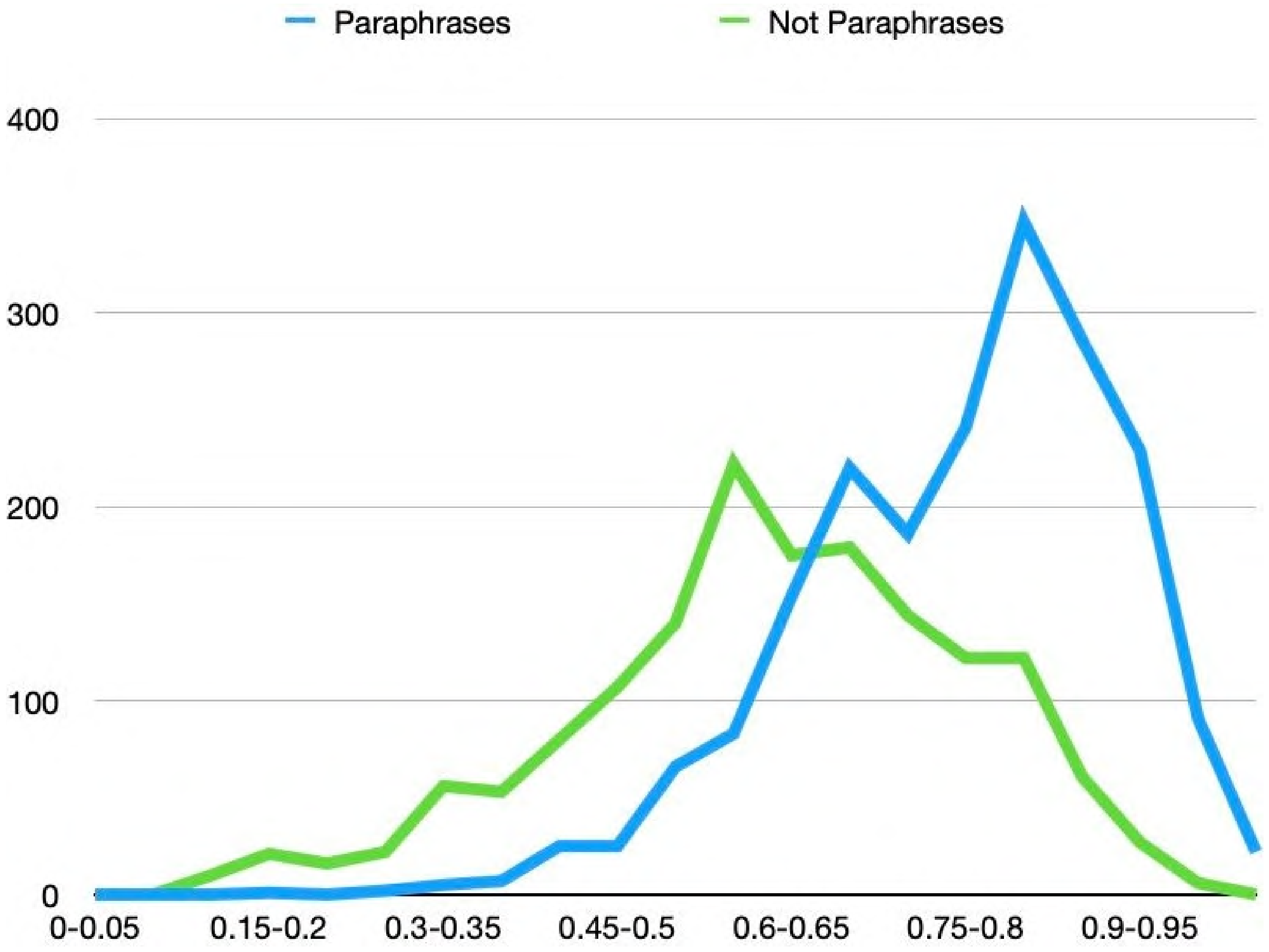
Case 1: As the chances of ambiguity is higher in sentences that are not highly similar, the probability distribution of noise is in a similar manner. Cosine similarity is thus, in a linear increasing fashion, from 90% in cosine similarity range 0-0.1, 80% in cosine similarity range 0.1-0.2 and so on.

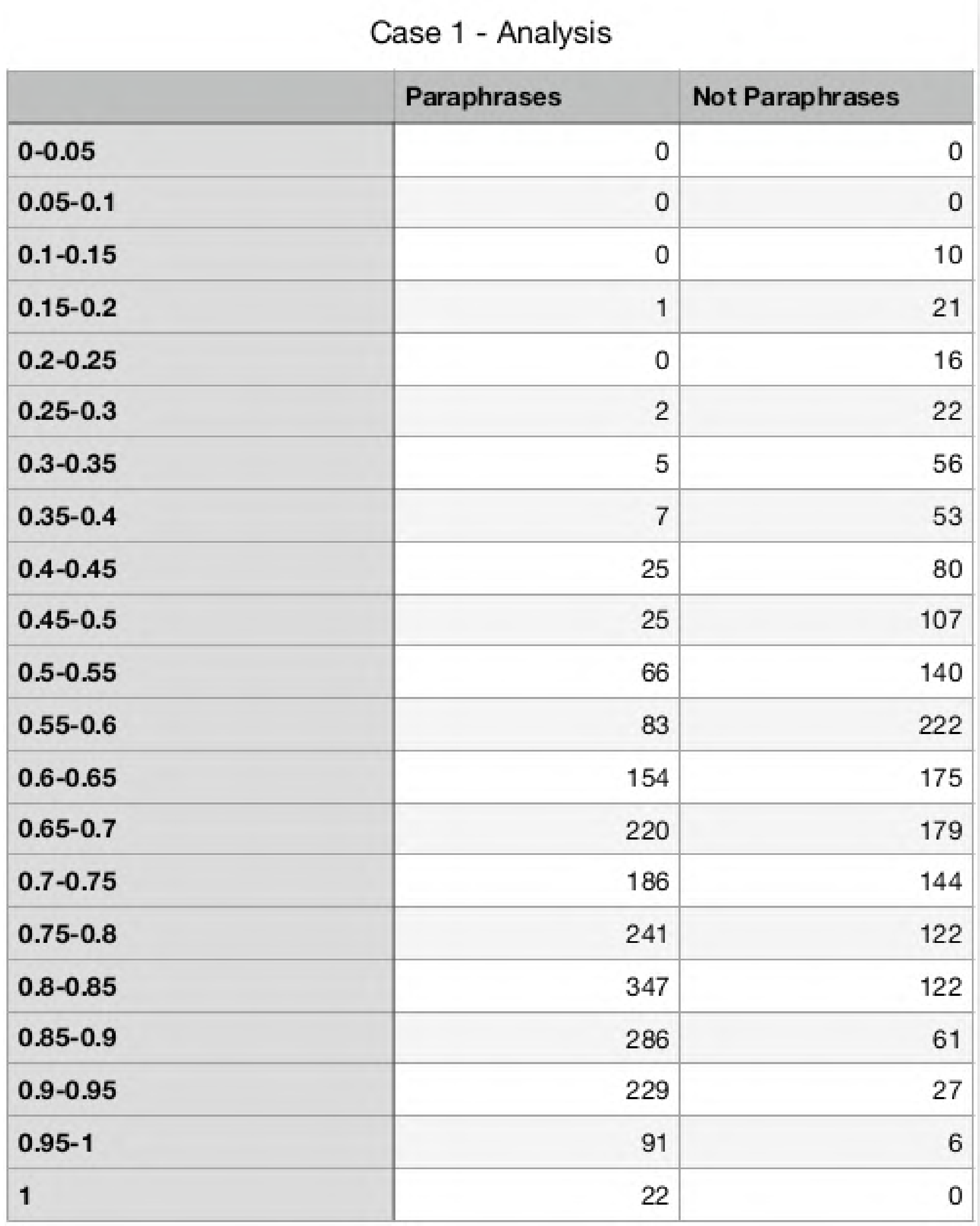
Optimum Threshold: 0.695 | Accuracy: 0.6689855072463768 | F1-score 0.7218704335119337





Analysis





Case 1 - Analysis

	Paraphrases	Not Paraphrases
0-0.05	0	0
0.05-0.1	0	0
0.1-0.15	0	10
0.15-0.2	1	21
0.2-0.25	0	16
0.25-0.3	2	22
0.3-0.35	5	56
0.35-0.4	7	53
0.4-0.45	25	80
0.45-0.5	25	107
0.5-0.55	66	140
0.55-0.6	83	222
0.6-0.65	154	175
0.65-0.7	220	179
0.7-0.75	186	144
0.75-0.8	241	122
0.8-0.85	347	122
0.85-0.9	286	61
0.9-0.95	229	27
0.95-1	91	6
1	22	0

Since the intersection lies in the range 0.65-0.7, therefore we can estimate the optimal to lie in the same range. The output 0.695 confirms this prediction.

Individual Contributions

PRAKHAR PANDEY 200101081

- Code Implementation: TF-IDF Paraphrase generation model
- Presentation Creation

PRANJAL BARANWAL 200101083

- Code Implementation: Noise induction + Preprocessing
- Ideation and Research on Paraphrase Generation Model

PRATHAM PEKAMWAR 200101087

- Statistical Analysis: Realistic Distribution
- Report Creation and Dataset Selection

DHRUV SHAH 200101124

- Statistical Interpretation : Random Distribution
- Ideation and Research on the Noise Induction Methods

Conclusion

For Random Distribution

We observe that the noise distribution does not affect the model's performance until the noise level are very high, on which the model breaks.

For Realistic Distribution (case1)

The performance of the model declines definitively in this case. This is because the optimal threshold increases to correspond to the distribution.

For Realistic Distribution (case2)

In this case the increase in label noise in label 1 and label 0 balance each other out, thus resulting in minimal change in optimal threshold and thus keeping the performance or accuracy unaffected.

Robustness

The TF-IDF model is quite robust when measured for noise sensitivity, however this comes in tradeoff to the accuracy, which is lower than neural network (BERT) models.