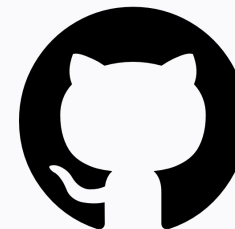


## Natural Language Processing with Disaster Tweets

# Predict which Tweets are about real disasters & which ones are not

**Son Nghiem**

Report + Coding



**Kien Nguyen**

Slides + Coding

Visit our Project

## Data Description

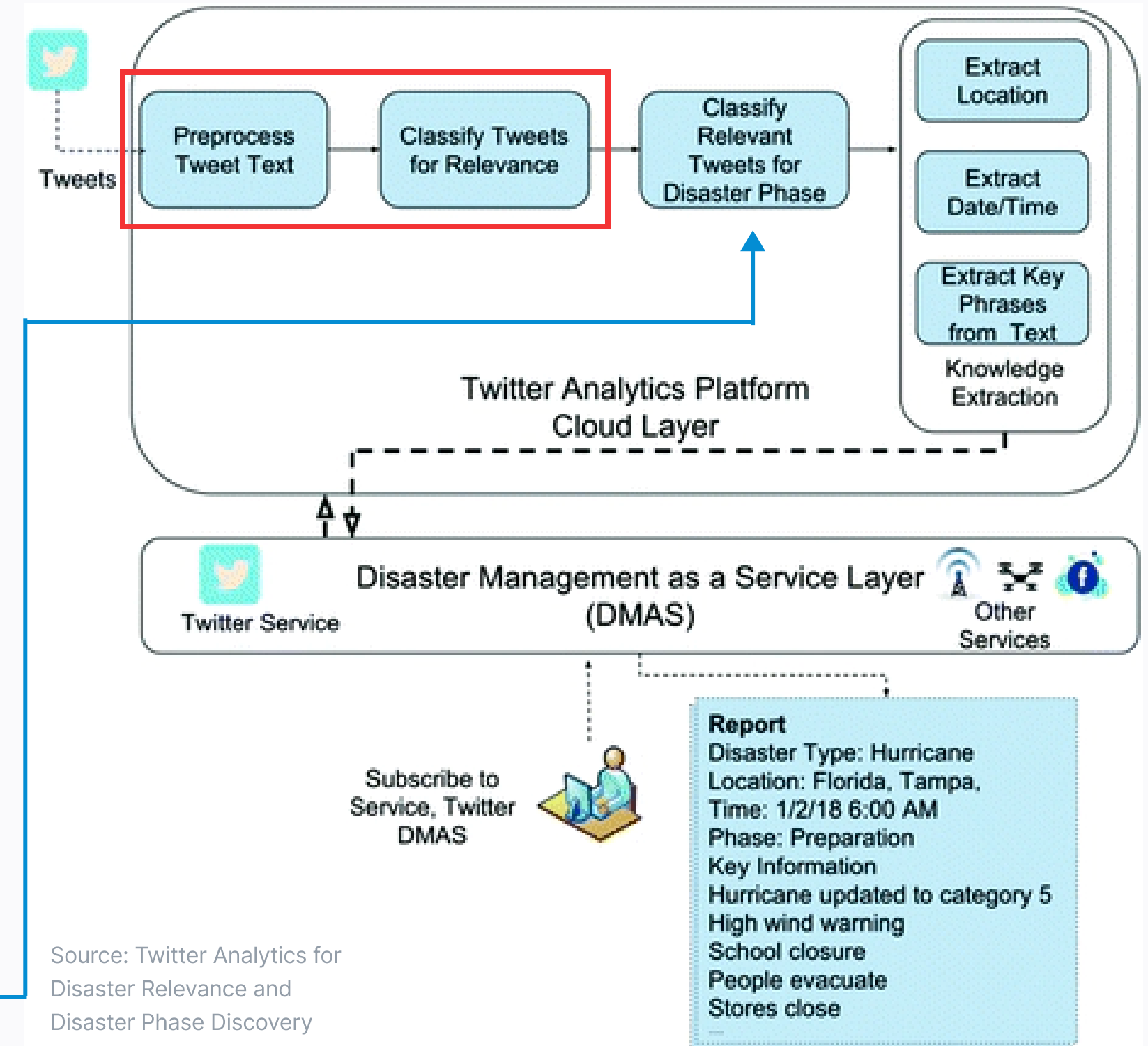
Size: ~11K Tweets  
(70% Training + 30% Test)

## Evaluation Metric

$$F_{\beta} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

Set **Beta** = 0.5

(Precision is *more important* than Recall)



## Features of Interest

### BEFORE

#### URL

Punctuation  
Hashtag  
Mention

count per  
Tweet

Most  
common

Hashtags  
Keywords  
Locations  
Named entity types

### AFTER

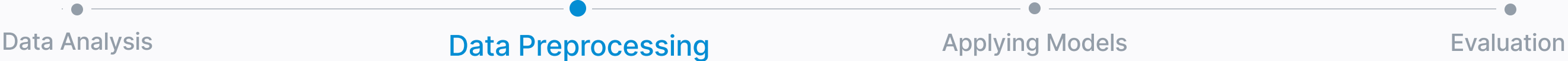
#### Character

Word  
Unique word  
Average word length

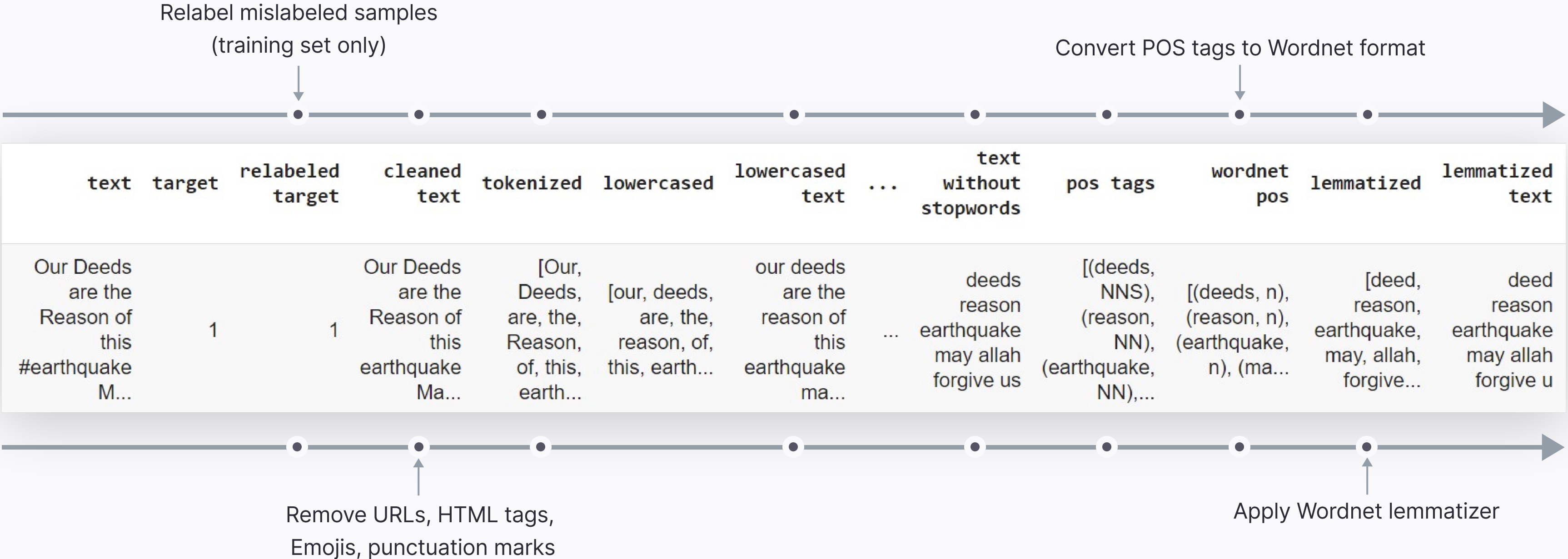
count per  
Tweet

Most  
common

Unigrams  
Bigrams  
Trigrams  
Fourgrams



Transformation



## Adding Potential Indicators

lemmatized text	url count	character count	bigram	disaster bigram count	non disaster bigram count
deed reason earthquake may allah forgive u	-0.934365	-0.492126	[allah forgive, deeds reason, earthquake may, ...	-0.483609	-0.960491

### From most common N-grams to Disaster/Non-Disaster N-gram count

*The greater the number  $N$ , the more disaster-relevant/irrelevant the most common  $N$ -grams in Disaster/Non-Disaster Tweets become, but also the less frequent they are.*

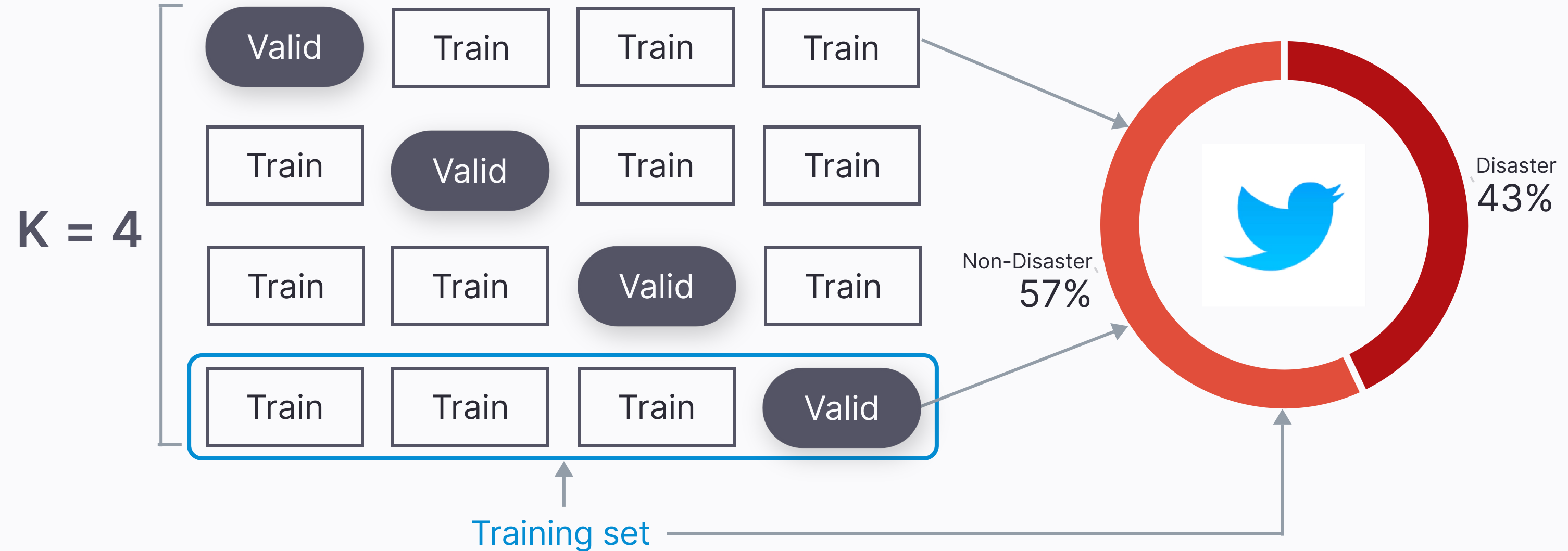
#### Problem:

- "Has any of the most common  $N$ -grams in Disaster Tweets" as potential indicator?  
⇒ For high  $N$ , many Disaster Tweets will be flagged as "No"

#### Solution:

- Create two lists of disaster-relevant/irrelevant  $N$ -grams from the most common  $[N+2]$ -grams in Disaster/Non-Disaster Tweets
- Count the number of  $N$ -grams of each Tweet in each of those lists

## Stratified Cross Validation





# Embedders

	Implementation	Vector Dimension
Bag-of-Words	<ul style="list-style-type: none"> <li>Use "sklearn.CountVectorizer" to embed Tweets directly</li> </ul>	Vocabulary length
GloVe	<ul style="list-style-type: none"> <li>Download the pretrained corpus model "glove.twitter.27B.50d.txt"</li> <li>Embed every word then Tweets by using the downloaded model</li> </ul>	50
Sentence-BERT	<ul style="list-style-type: none"> <li>Continue training the fine-tuned Sentence Transformer model "paraphrase-MiniLM-L6-v2" on pairs of Disaster and/or Non-Disaster Tweets</li> <li>Use the continue-trained model to embed Tweets directly</li> </ul>	384
TF-IDF	<ul style="list-style-type: none"> <li>Use "sklearn.TfidfVectorizer" to embed Tweets directly</li> </ul>	Vocabulary length
Word2Vec	<ul style="list-style-type: none"> <li>Download the pretrained corpus model "word2vec-google-news-300"</li> <li>Embed every word then Tweets by using the downloaded model</li> </ul>	300

Data Analysis

Data Preprocessing

Applying Models

Evaluation

## Classifiers

Logistic  
Regression

**L2** Regularization

Multilayer  
Perceptron

**ReLU** activation

**One** hidden layer with **100** neurons

Max #Iterations **200**

**Constant** learning rate **0.001**

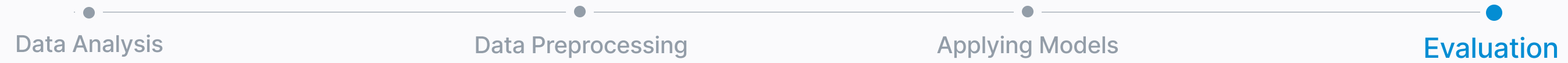
Random  
Forest

**100** Decision Trees

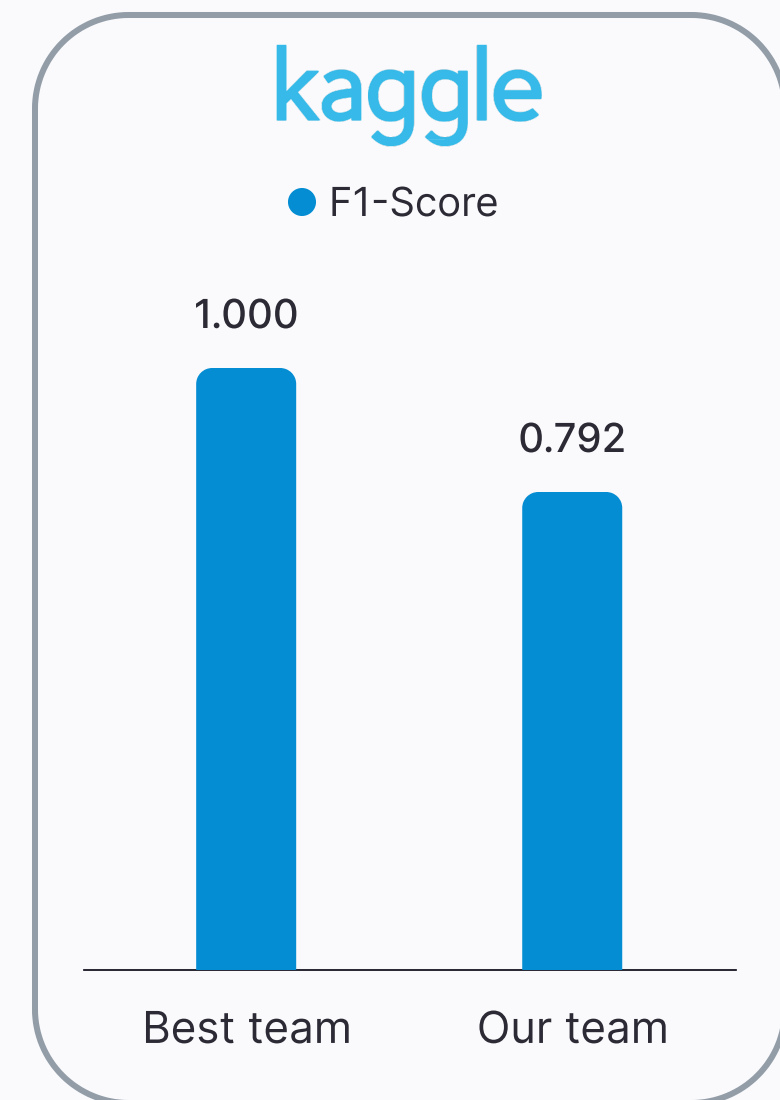
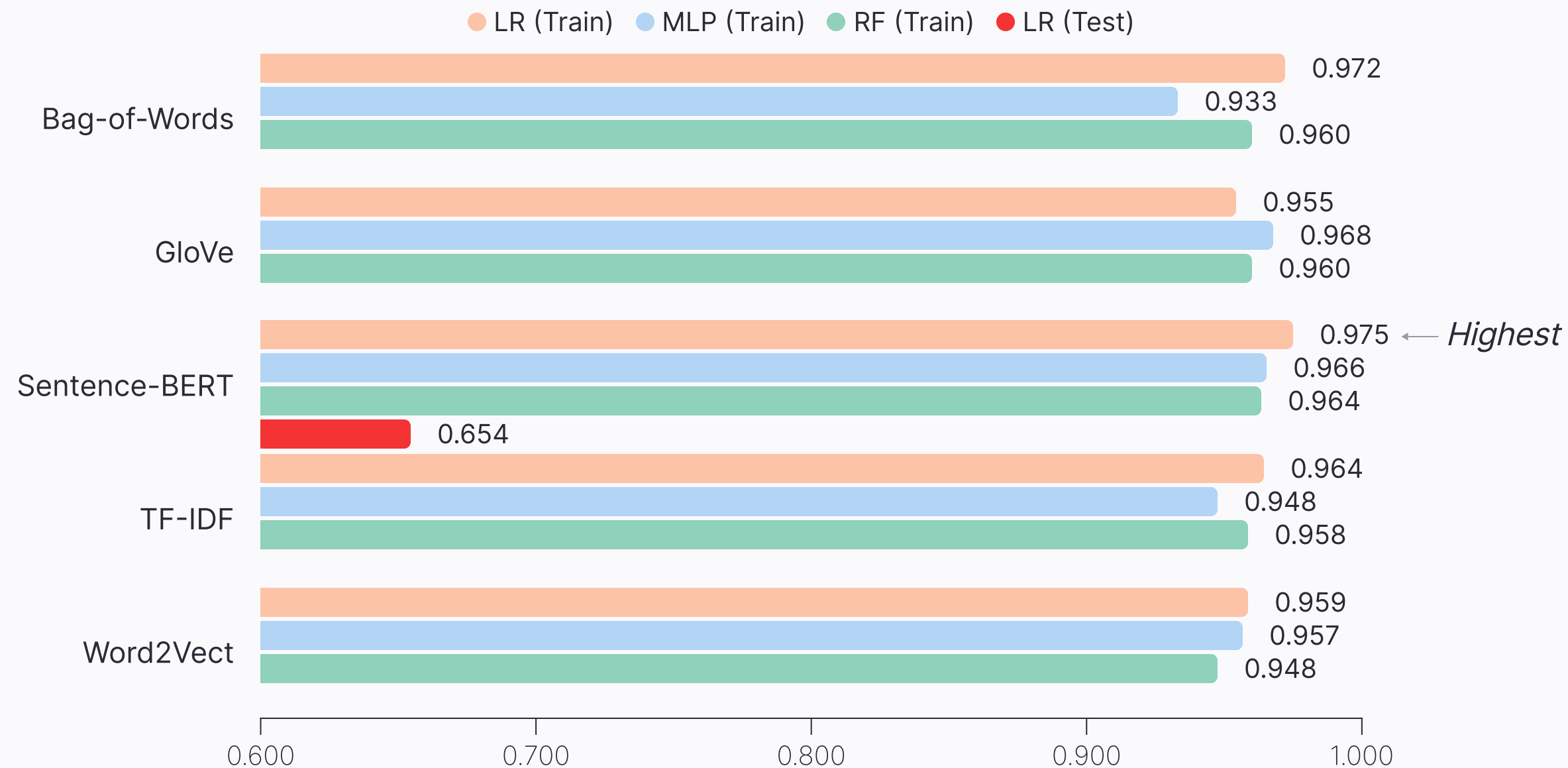
**Gini Impurity** as split criteria

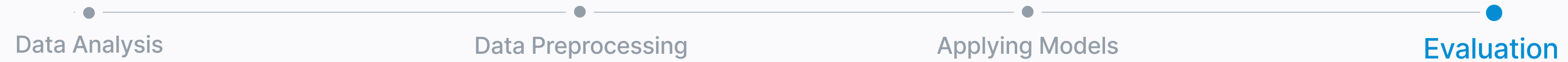
Build trees **with Bootstrapping**





## Results





## Discussion

### PROBLEM

### OPTIMIZATION

Human error during labeling test set

Label test set with a gold standard model

No hyperparameter tuning  
for Logistic Regression Classifier

Hyperparameter Optimization  
with Random Search & Grid Search

Only one training epoch  
for Sentence-BERT Embedder

Increase the number of training epochs  
for Sentence-BERT Embedder

Inaccurate & incomplete training samples  
for Sentence-BERT model

Evaluate samples systematically  
before training Sentence-BERT model

Possibility of added potential indicators  
being insignificant

Treat potential indicator choices  
as hyperparameters during fine-tuning