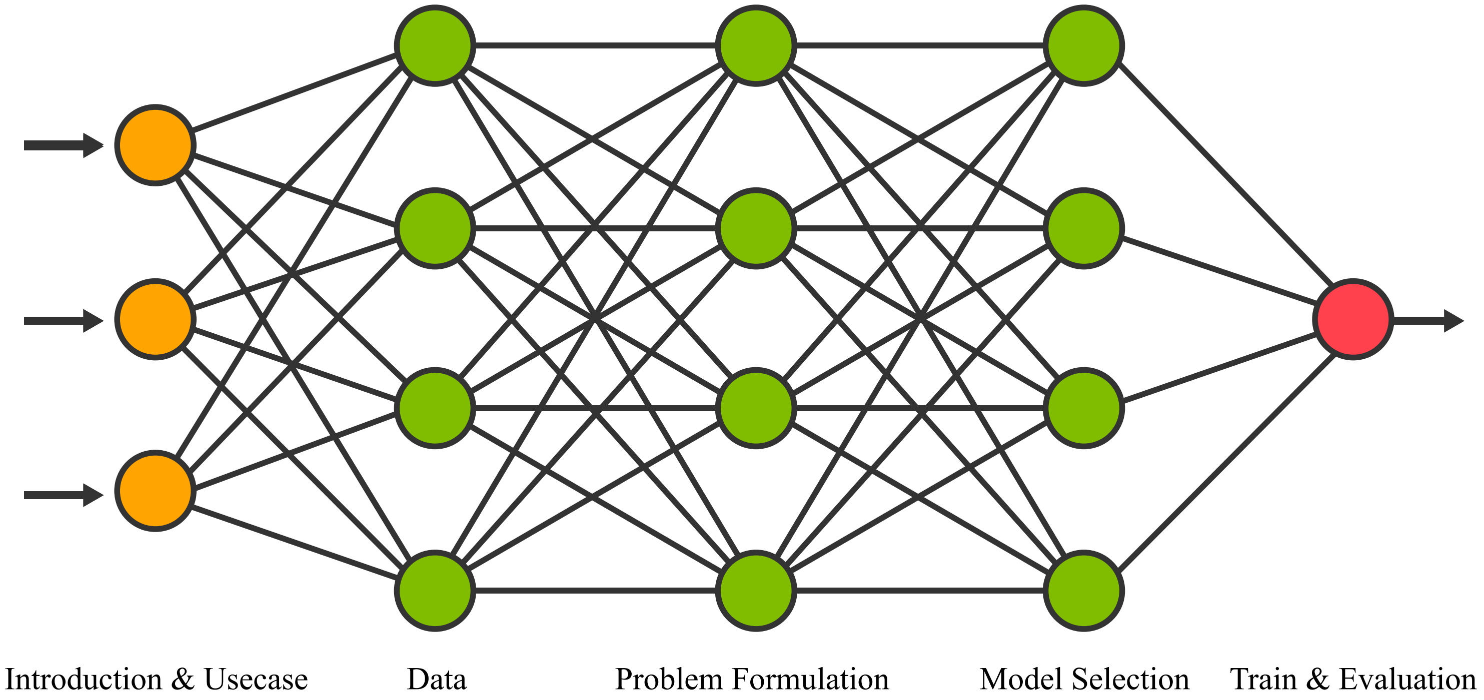
Disaster Detection System

Alireza Heidari

Fall 2023


Introduction & Usecase
Data
Problem Formulation
Model Selection
Train & Evaluation

Introduction

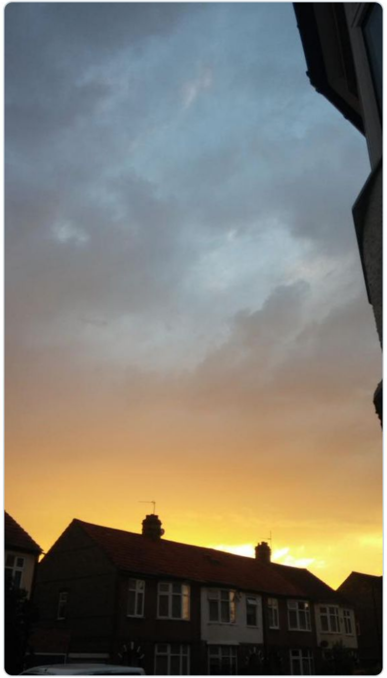
Our goal is to have a system that predicts which Tweets are about real disasters and which one's aren't.

Only through access to a medium-sized dataset of Tweets can this goal be achieved.

The author explicitly uses the word “ABLAZE” but means it metaphorically. This is clear to a human right away, especially with the visual aid. But it’s less clear to a machine.

Anna K
@AnyOtherAnnaK

On plus side LOOK AT THE SKY LAST NIGHT IT WAS ABLAZE



12:43 AM · Aug 6, 2015 · Twitter for Android

Usecases and Real-world Applications

Prevent bad things or accidents: By identifying tweets that are mentioning real disasters, our model could be used to alert first responders and other relevant authorities. This could help to prevent bad things from happening, or to minimize the damage that is caused.

Provide early warning of disastrous events: By identifying tweets that are mentioning real disasters, our model could be used to provide early warning of these events. This could give people time to prepare for the event, or to evacuate to safety.

Improve the accuracy of disaster reporting: By identifying tweets that are mentioning real disasters, our model could be used to improve the accuracy of disaster reporting. This could help to ensure that people are getting accurate information about disasters, and that they are not being misled by false or inaccurate reports.

Data

10,000+ tweets related to disaster keywords (e.g., "crash", "quarantine", "bush fires") with associated location and keyword.

Collection date: Jan 14th, 2020.

	id	keyword	location	text	target
31	48	ablaze	Birmingham	@bbcmtd Wholesale Markets ablaze http://t.co/l...	1
32	49	ablaze	Est. September 2012 - Bristol	We always try to bring the heavy. #metal #RT h...	0

Data Quality Assessment - Missing Values

Location:

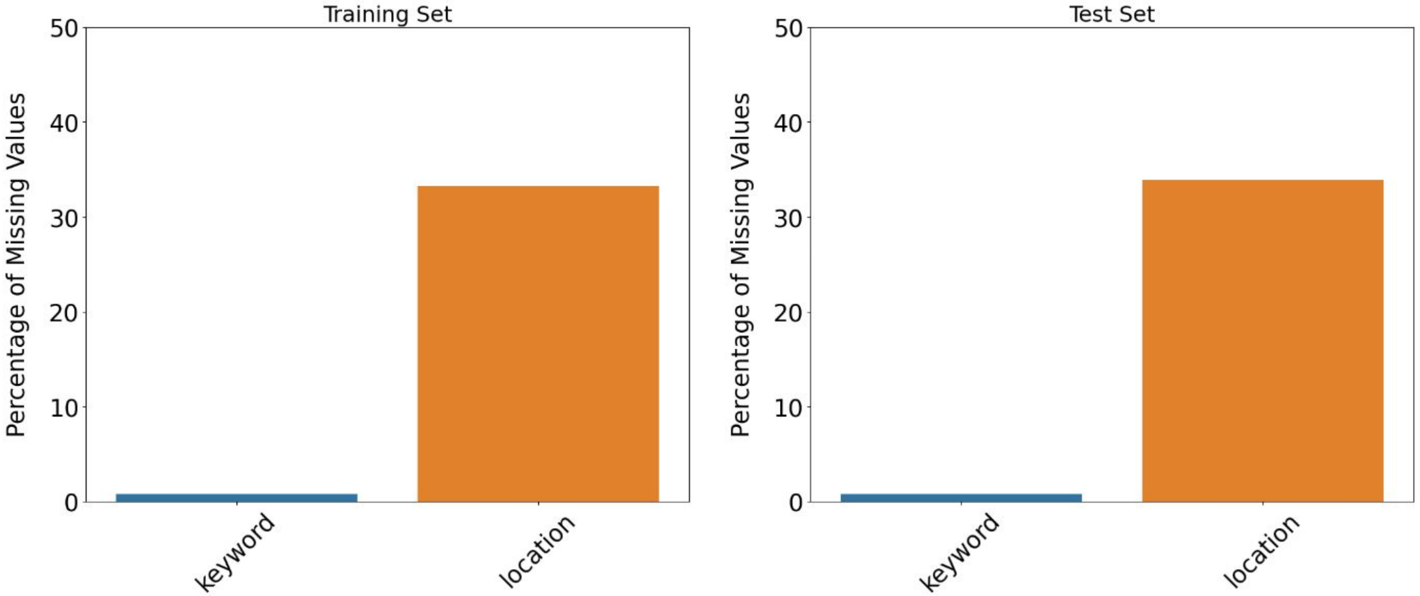
- A 33% missing value ratio in both the training and test sets. Missing values are imputed with **no_location**.

Keyword:

- A 0.8% missing value ratio in both training and test sets. Missing values are imputed as **no_keyword.**

Since missing value ratios between training and test set are too close, t**hey are most probably taken from the same sample**.

The decision to employ default values for imputation stems from the inherent noise present in tweets. Applying interpolation or extraction methods may potentially introduce unintended distortions to the data quality.


Training Set
Test Set
Percentage of Missing Values
50
40
30
20
10
0
keyword
location

Data Quality Assessment - Feature Reduction

The location feature is not automatically generated, but is instead a user input. This makes it a very noisy feature, with too many unique values to be useful as a feature.

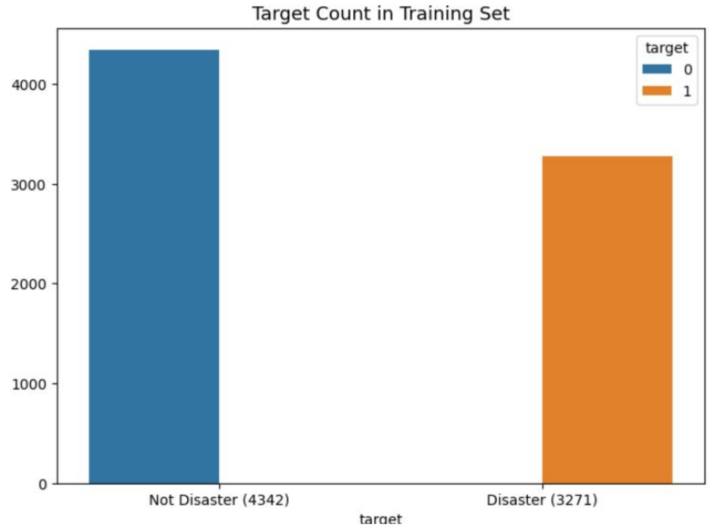
Data Exploration

Methods:

- Target Distribution
- Correlation Between Keyword and Target Label
- Exploring N-Grams and The Relation with the Target Label

Target Distribution

The class distribution for the target variable is **57% for 0** (Not Disaster) and **43% for 1** (Disaster). The classes are almost equally balanced, so stratification by target is not necessary during cross-validation.

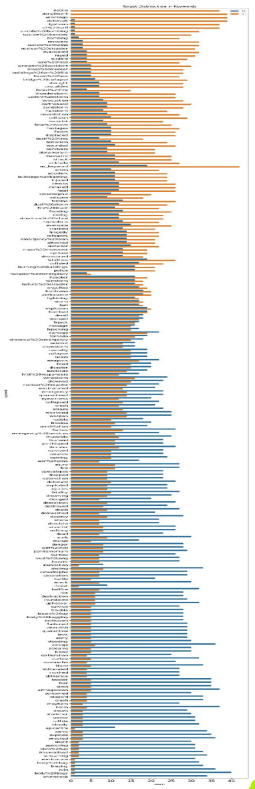


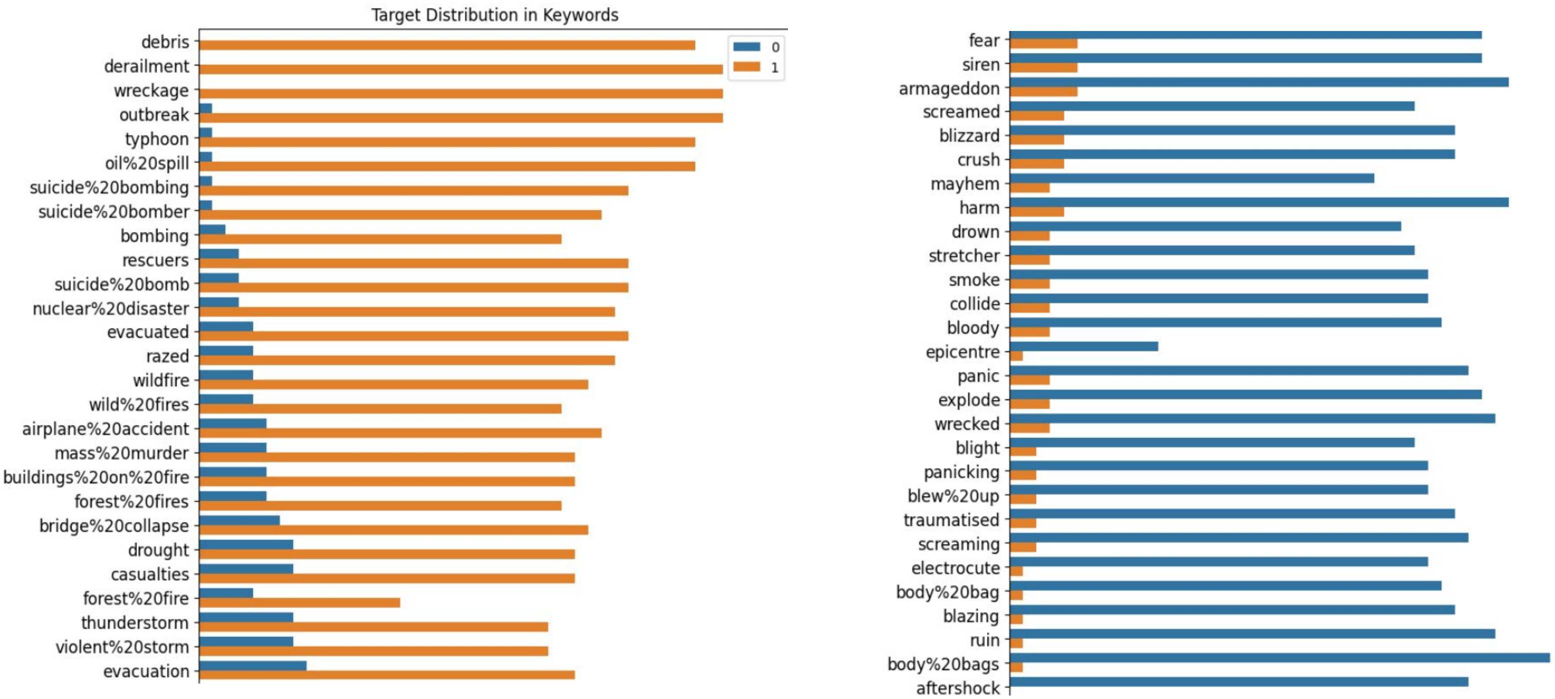
Correlation Between Keyword and Target Label

222 unique keywords.

We plot the number of occurrence of each keyword with respect to the corresponding target.

The observed figure illustrates a notable and statistically significant correlation between the keywords and the target variable.

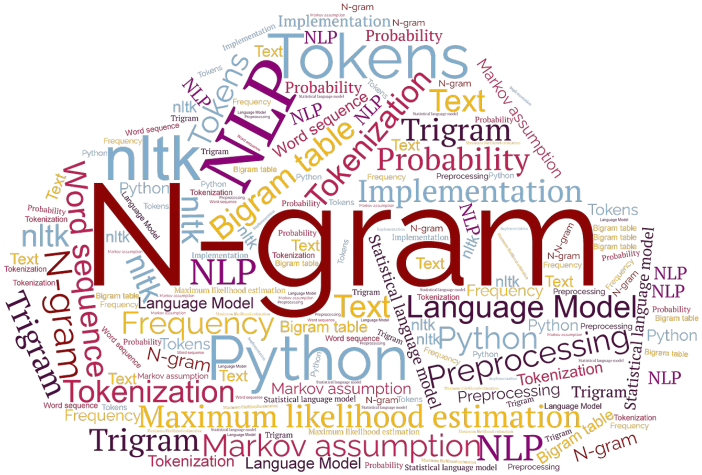



Target Distribution in Keywords
0
1
debris
derailment
wreckage
outbreak
typhoon
oil%20spill
suicide%20bombing
suicide%20bomber
bombing
rescuers
suicide%20bomb
nuclear%20disaster
evacuated
razed
wildfire
wild%20fires
airplane%20accident
mass%20murder
buildings%20on%20fire
forest%20fires
bridge%20collapse
drought
casualties
forest%20fire
thunderstorm
violent%20storm
evacuation
fear
siren
armageddon
screamed
blizzard
crush
mayhem
harm
drown
stretcher
smoke
collide
bloody
epicentre
panic
explode
wrecked
blight
panicking
blew%20up
traumatised
screaming
electrocute
body%20bag
blazing
ruin
body%20bags
aftershock

Exploring N-Grams and The Relation with the Target Label

Bigrams

Trigrams



Bigrams

Shared Bigrams in Both Classes

- None observed in both classes due to distinct contextual cues.

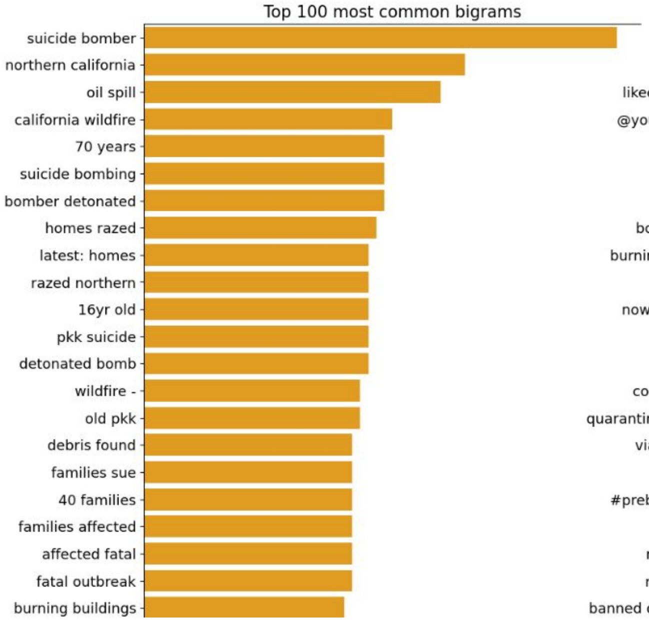
Bigrams in Disaster Tweets

- Offer enhanced disaster information compared to unigrams.
- Punctuation removal required for accurate analysis.

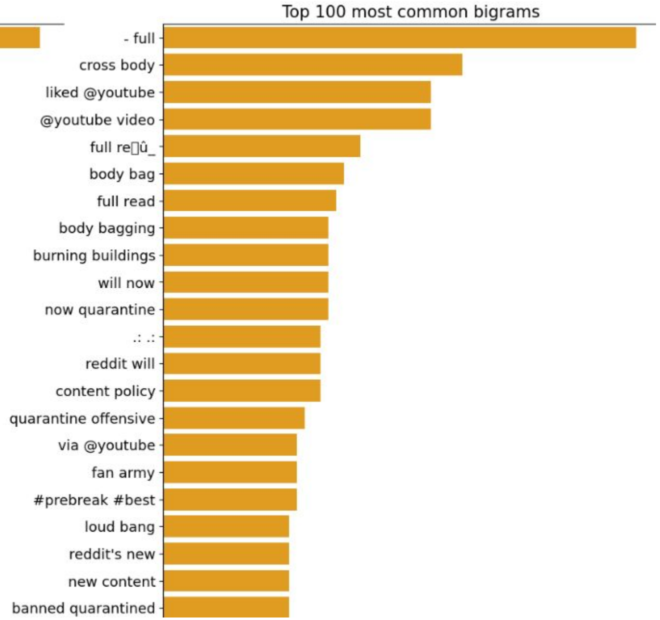
Unigrams in Non-Disaster Tweets

- Frequently related to Reddit or YouTube.
- Contains substantial punctuation, necessitating cleaning.

Disaster Tweets



Non-Disaster Tweets



Trigrams

Shared Trigrams in Both Classes

- None found in both classes due to distinct contextual cues.

Trigrams in Disaster Tweets

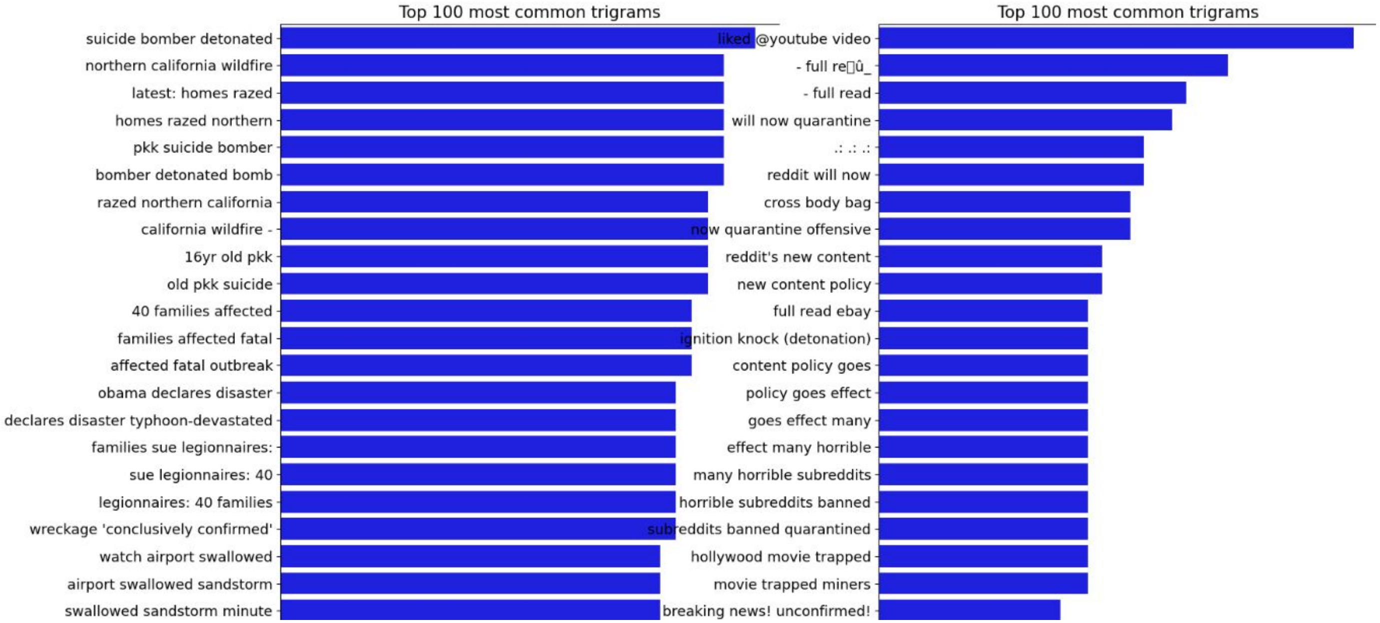
- Resemble bigrams, offering substantial disaster information.
- May not always provide additional insights beyond bigrams.

Trigrams in Non-Disaster Tweets

- Similar to bigrams, with increased punctuation usage.

Disaster Tweets

Non-Disaster Tweets

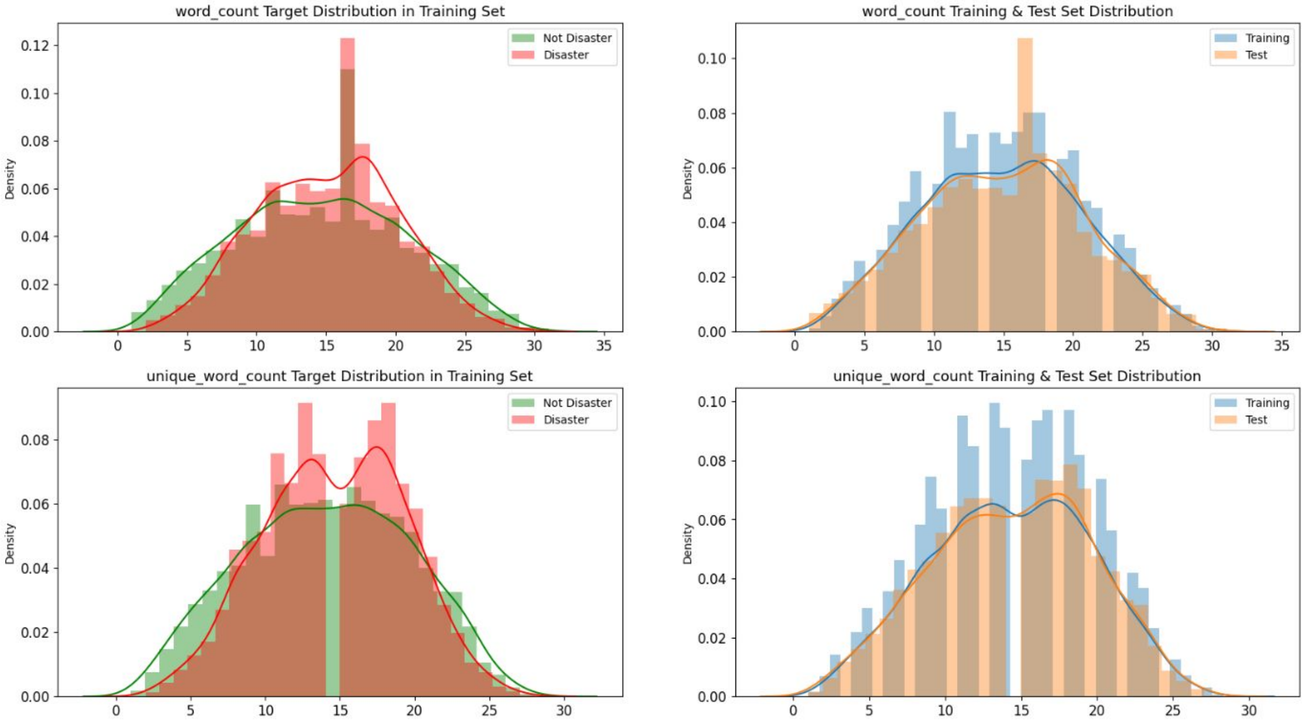


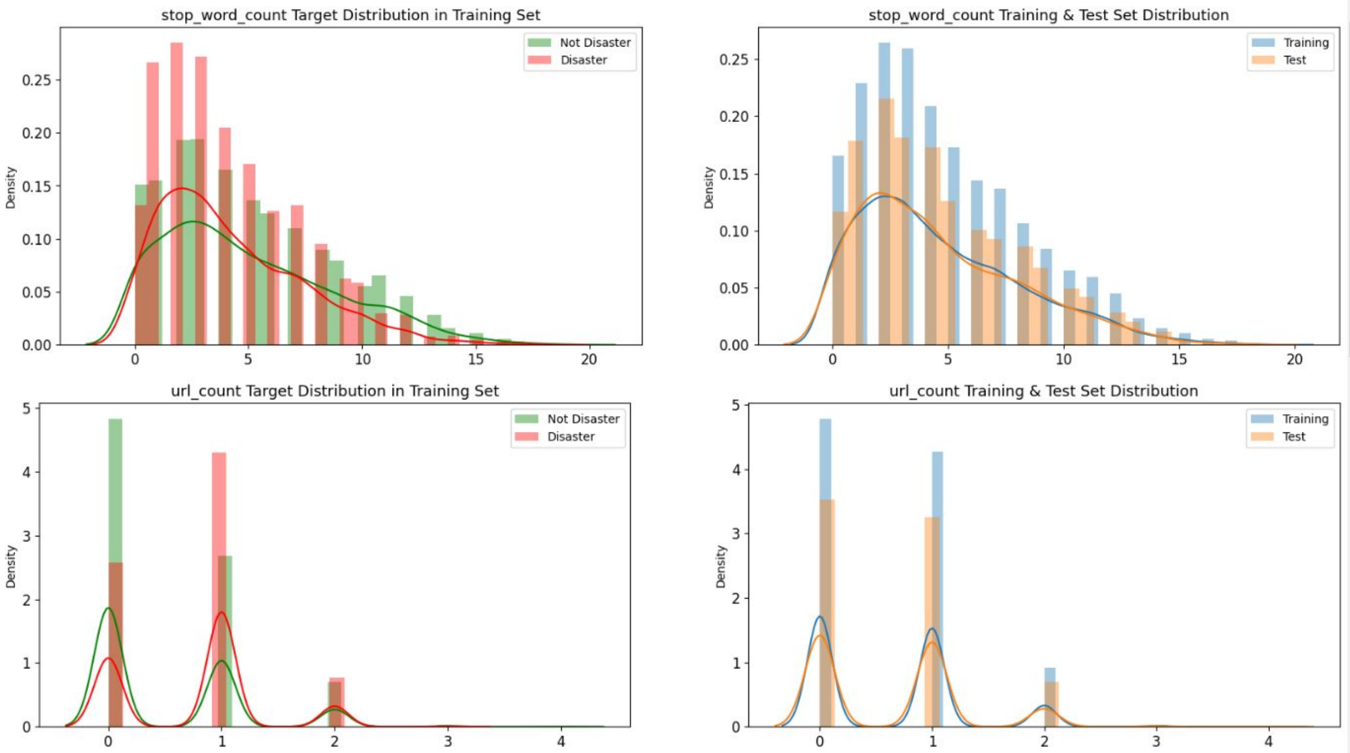
Feature Engineering

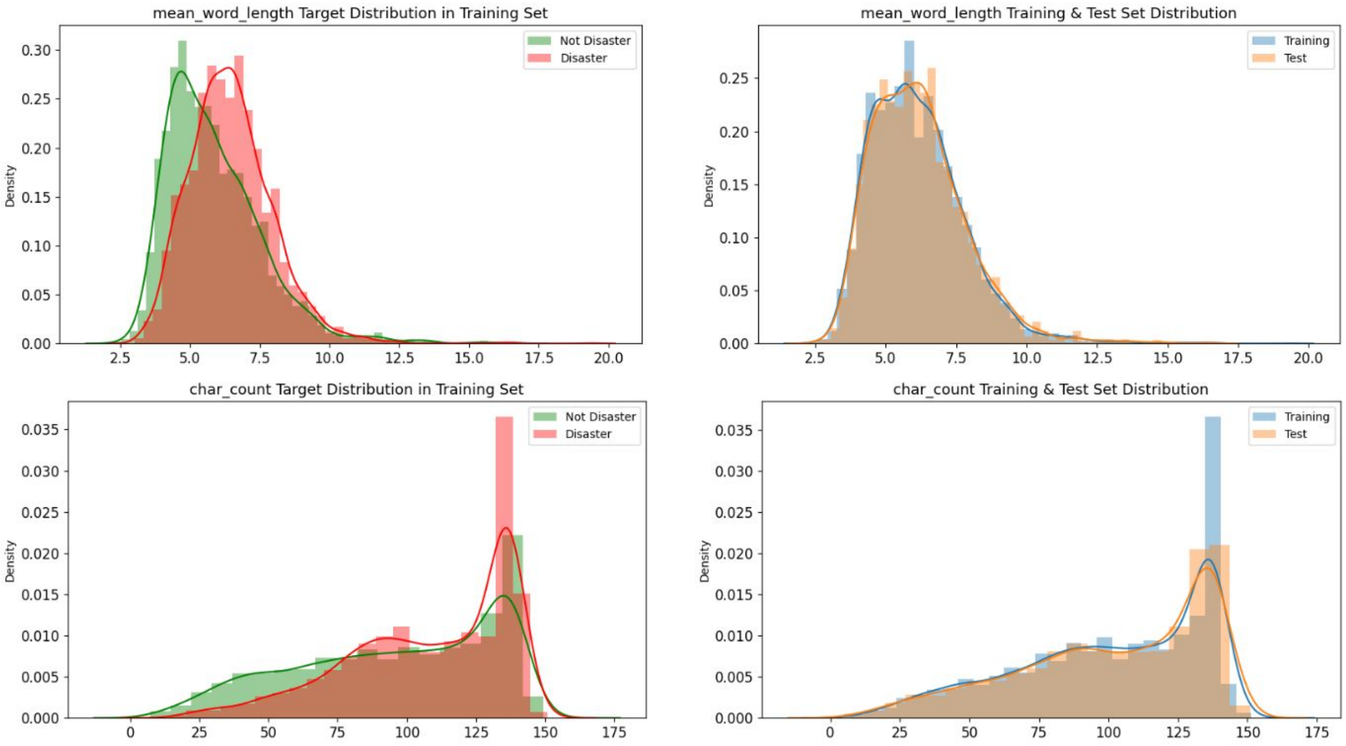
Understanding the linguistic features within classes and datasets aids in discerning disaster tweets. These distinctions arise from factors such as formality, word length, and source.

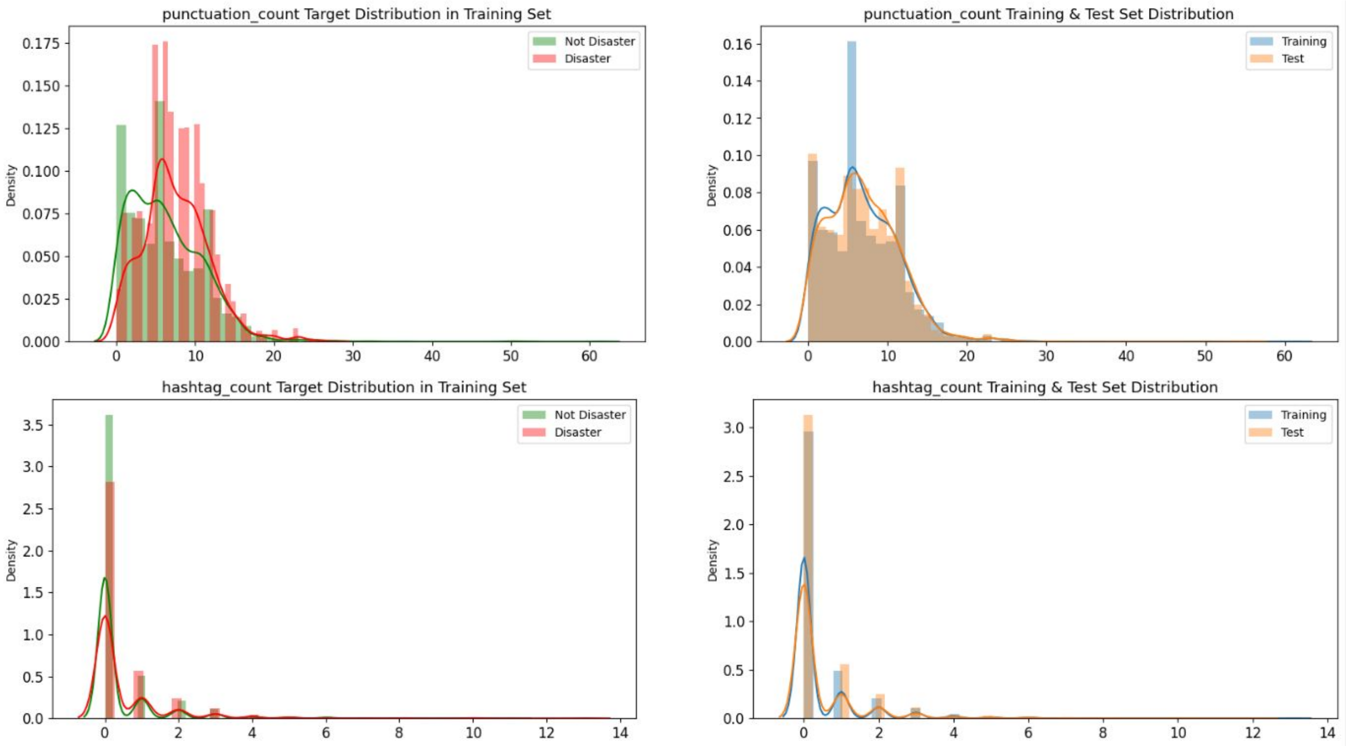
Key Linguistic Features:

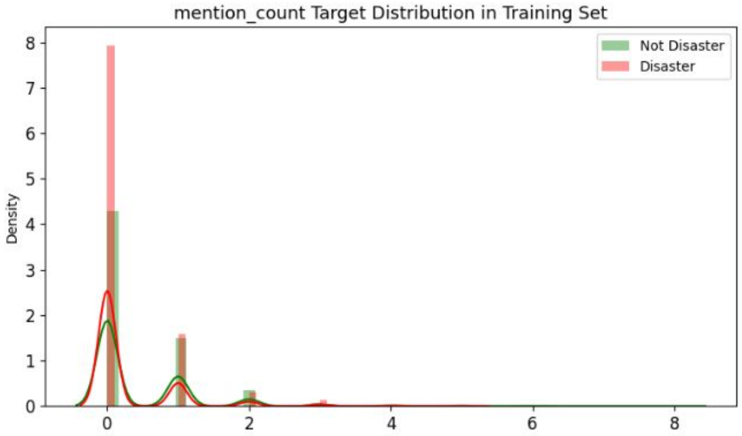
- **word_count:** Total word count in the text.
- **unique_word_count:** Count of unique words in the text.
- **stop_word_count:** Number of stop words in the text.
- **url_count:** Occurrences of URLs in the text.
- **mean_word_length:** Average character count per word.
- **char_count:** Total character count in the text.
- **punctuation_count:** Total punctuation marks in the text.
- **hashtag_count:** Number of hashtags (#) in the text.
- **mention_count:** Number of mentions (@) in the text.

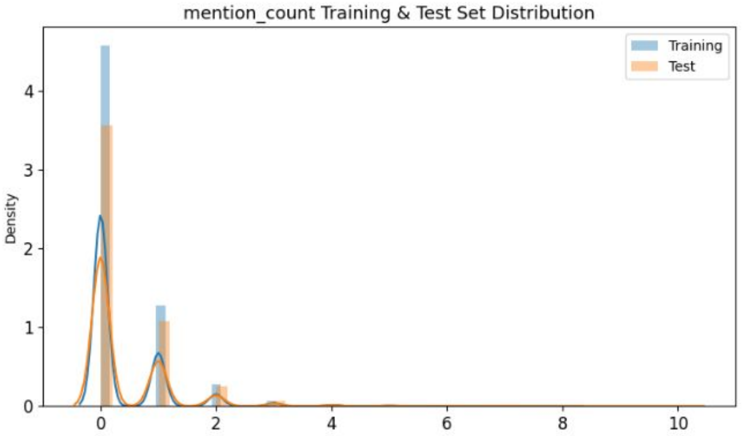

word_count Target Distribution in Training Set
Not Disaster
Disaster
Density
word_count Training & Test Set Distribution
Training
Test
Density
unique_word_count Target Distribution in Training Set
Not Disaster
Disaster
Density
unique_word_count Training & Test Set Distribution
Training
Test
Density


stop_word_count Target Distribution in Training Set
Not Disaster
Disaster
Density
stop_word_count Training & Test Set Distribution
Training
Test
Density
url_count Target Distribution in Training Set
Not Disaster
Disaster
Density
url_count Training & Test Set Distribution
Training
Test
Density


mean_word_length Target Distribution in Training Set
Not Disaster
Disaster
Density
mean_word_length Training & Test Set Distribution
Training
Test
Density
char_count Target Distribution in Training Set
Not Disaster
Disaster
Density
char_count Training & Test Set Distribution
Training
Test
Density


punctuation_count Target Distribution in Training Set
Not Disaster
Disaster
Density
punctuation_count Training & Test Set Distribution
Training
Test
Density
hashtag_count Target Distribution in Training Set
Not Disaster
Disaster
Density
hashtag_count Training & Test Set Distribution
Training
Test
Density

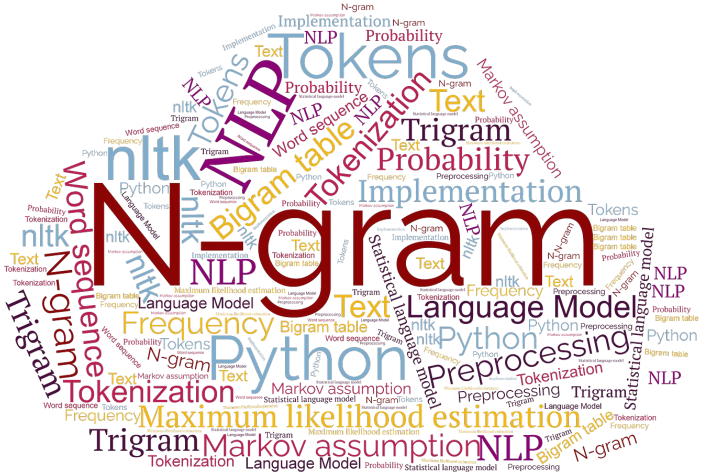

mention_count Target Distribution in Training Set
Not Disaster
Disaster
Density
0
1
2
3
4
5
6
7
8
0
2
4
6
8


mention_count Training & Test Set Distribution
Training
Test
Density
0
1
2
3
4
0
2
4
6
8
10

Exploring N-Grams and The Relation with the Target Label

Bigrams

Trigrams



Data Cleaning

Efficient cleaning of tweets is essential for meaningful analysis. However, manual cleaning of individual tweets is impractical due to time constraints, necessitating a systematic approach.

General Cleaning Approach

- Removing special characters attached to words.
- Expanding contractions.
- Eliminating URLs.
- Substituting character entity references.
- Correcting typos, slang, and informal abbreviations.
- Replacing words with acronyms.
- Grouping certain words.

Problem Formulation

1. **Classification Task -> Cross Entropy Loss** $L = -\frac{1}{m}\sum_{i=1}^{m} y_i \cdot \log(\hat{y}_i)$
2. Model Evaluation:
   a. Accuracy
   b. Precision
   c. Recall
   d. F1-Score

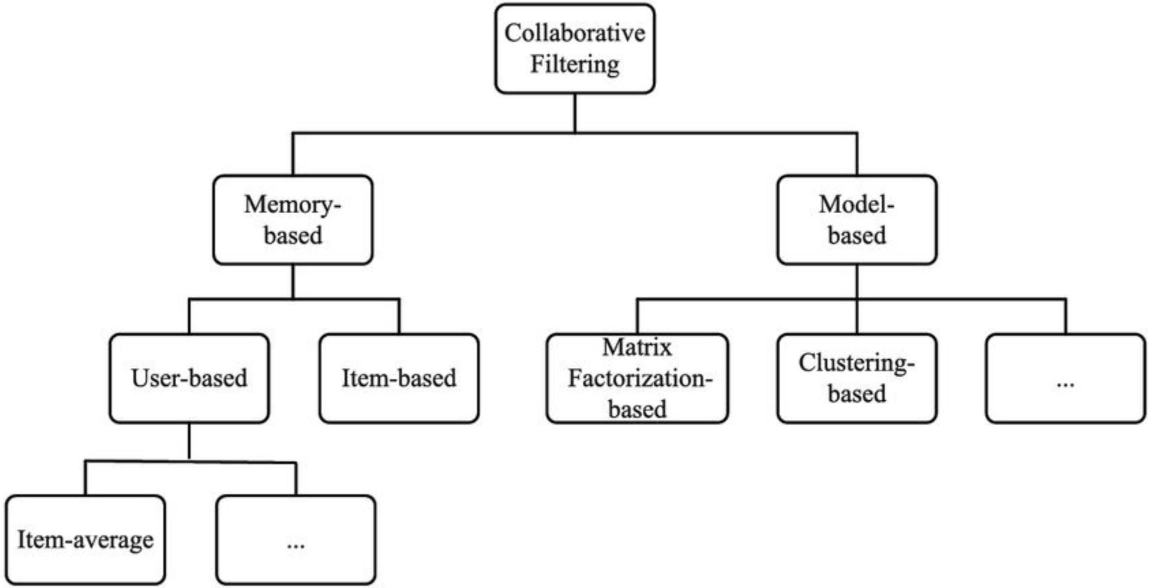
Due to minor class-imbalance, accuracy, recall, and precision may not fully reflect model performance on both classes. Hence, we prioritize **F1-score as the primary metric**.

Model Selection

Two approaches:

- Traditional ML models -> Naive Bayes
- Deep Learning algorithms -> LLMs (BERT)

Model Selection



Model Selection

Two approaches:

- Traditional ML models -> Naive Bayes
- Deep Learning algorithms -> LLMs (BERT)

Naive Bayes

Despite its simplicity, it excels in text classification tasks like spam detection.

It assumes features are independent.

Likelihood of the Evidence given that the Hypothesis is True

Prior Probability of the Hypothesis

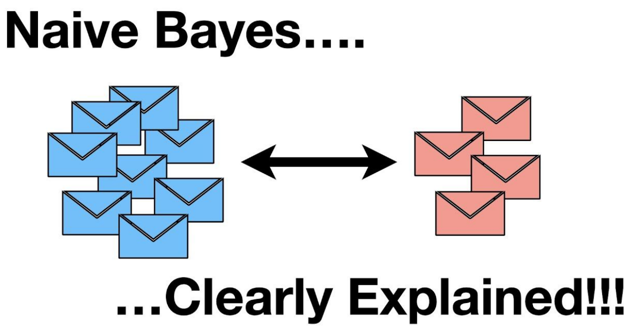
$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)}$$

Posterior Probability of the Hypothesis given that the Evidence is True

Prior Probability that the evidence is True

Real-world Applications:

- Medical Diagnosis and Treatment Planning
- Fraud Detection
- Natural Language Processing
- Environmental Modeling
- Fault Diagnosis in Engineering

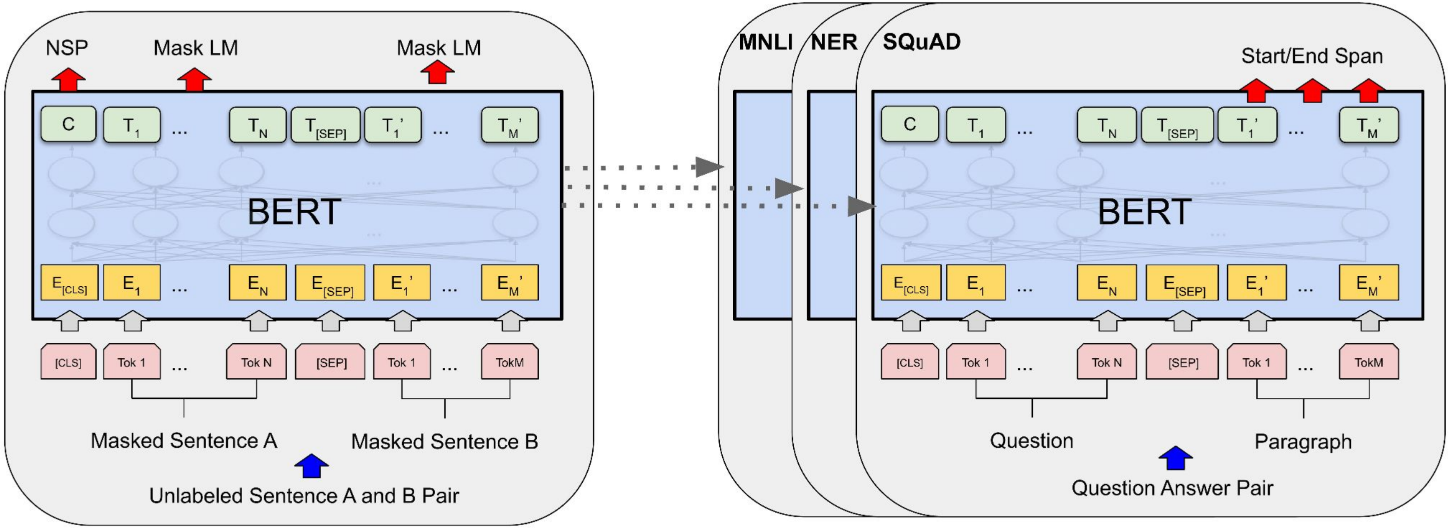


BERT (Bidirectional Encoder Representations from Transformers)

A pre-trained deep learning model for natural language processing tasks. It revolutionized NLP by considering context from both left to right and right to left, capturing intricate language nuances.

Training Data: BERT is trained on an extensive corpus of text data, encompassing diverse sources.

Training Objective: It learns to predict missing words in sentences, enabling it to grasp contextual nuances.



Pre-training

Fine-Tuning

Real-world Applications:

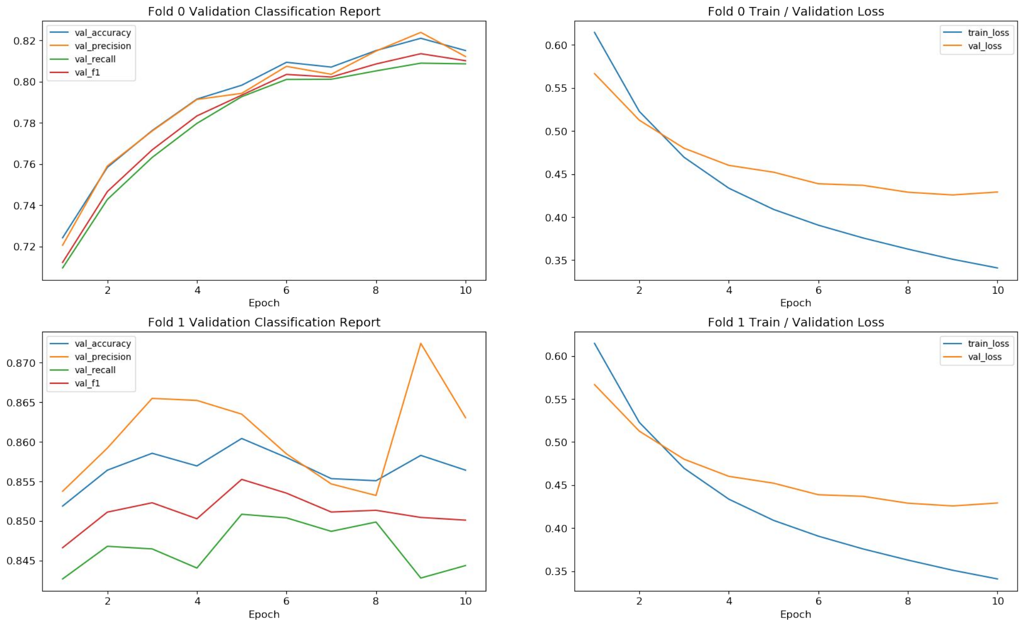
- **Question Answering:** BERT powers QA systems by understanding the context around questions.
- **Sentiment Analysis:** It excels in sentiment analysis tasks by capturing intricate sentiment expressions.
- **Named Entity Recognition:** BERT helps in accurately identifying entities in text

Training

Training Configs:

- Naive Bayes:
    - TF vectorizing (count)
- BERT:
    - Tokenization through the BERT’s tokenizer
    - SGD optimizer (lr=0.0001)
    - 10 epochs
    - Batch size of 32
    - Maximum sequence length of 128

BERT Training Monitoring



Evaluation

Model	Precision	Recall	Accuracy	F1-Score
BERT	86% **(+4%)**	84% **(+14%)**	85% **(+29%)**	**86% (+11%)**
Naive Bayes	82%	70%	56%	**75%**

Evaluation

As demonstrated earlier, the BERT model exhibited a remarkable performance advantage over Naive Bayes. Therefore, **we select the fine-tuned BERT model as our chosen approach**.

Conclusion

We believe that our model has the potential to make a significant impact on the world. By helping to prevent bad things or accidents, and by providing early warning of disastrous events, our model could help to save lives and reduce suffering. We are committed to continuing our work on this project, and to making our model available to those who can use it to make the world a safer place :)

I hope you have found this presentation informative.