An NLP model for Sarcasm Detection

Problem Statement

Develop an NLP classification model for sarcasm detection using the features provided in the dataset. During the course, explore NLP concepts and models. Further, evaluate and finalize the best modeling approach for the given dataset.

Problem Overview

Sarcasm is the caustic use of words, often in a humorous way, to mock someone or something. Sarcasm is a nice trait to have. However, there is a thin line between sarcasm and foul language. You have to define that thin line by building a classic NLP classification model using the provided dataset.

Metric

Evaluate your models on F1_score: which combines the precision and recall of a classifier into a single metric by taking their harmonic mean.

Dataset Details

A university has chat groups on different topic. Students & their parents both have access to these chat groups. The dataset contains chat extract from the chat groups along with topic name and few other parameters, out of which three parameters description are classified (not disclosed).

This dataset contains following independent features.

1. **ID**: id of student
2. **comment** : Student sarcastic comment
3. **date:** Date on which comment was recorded
4. **down :** Undisclosed parameter
5. **parent comment :** Parent comment on the same topic
6. **score** : Undisclosed/classified parameter
7. **top** : Undisclosed/classified parameter
8. **topic** : Topic of the discussion
9. **user** : Chat login name of the student
10. **label** : Sarcasm level (0 -not a sarcasm , 1 -is sarcasm)

Dataset Cleaning

Explore the student comment & parent comment features by creating a text corpus. Which all cleaning operation you think will be required on this corpus? Write a clean-up method and clean the text features.

Explore Classics ML models for your NLP model

Perform text to numeric conversion using CountVectorization and also with TF-IDF on the cleaned dataset. Now process the whole DataFrame (vectorize text + other features) with classic ML models for example- Logistic Regression, Naive Bayes, LDA, Decision tree etc. Tune you models and suggest what combination of vectorization technique & ML model is most suitable for the given data set.

Explore Word Embeddings & classic DL models

Preform text vectorization using word embedding techniques (Word2Vec & Glove). Now use the embedding to build DL models such as RNN, LSTM & Bi-LSTM. Tune you models and suggest what combination of embedding technique and DL model is most suitable for the given data set. Now for the finalized embedding method use its pretrained word embeddings and test if model performance can be improved.

Explore State-of the Art Transformer models

Use any two state-of-the-art transformer models and check if you can improve NLP model performance further.

Submission Deliverables

1. The complete script(s) should be submitted in either one or multiple .ipynb (or html) format files with all required processing & outputs.

2. PPT deck (8-10 slides) to brief about your model's -execution, comparative results and final conclusion.