Recognition of Sarcasm in Tweets Based on Concept Level Sentiment Analysis and Supervised Learning Approaches

INFORMATION RETRIEVAL
PROJECT REPORT

GROUP MEMBERS:

AREEB UL HAQ	21K-3346
HAMOOD SIDDIQUI	21K-4539
YAHYA HUSSAIN	21K-4895

COURSE INSTRUCTOR:
SIR BASIT JASANI

Project Overview

The project focuses on developing a machine learning model to detect sarcasm in headlines using natural language processing (NLP) techniques. The model is trained on a dataset of headlines, incorporating various NLP-based features to improve its accuracy. The project is structured into several modules, each responsible for different tasks in the data preprocessing and model training pipeline.

Data Preparation

1. Loading and Cleaning Data:
    - The dataset is loaded from a JSON file, and unnecessary columns, such as article links, are removed.
    - Headlines are lemmatized using the spaCy library to reduce words to their base forms, which helps in normalizing the text for analysis.
2. Idioms Handling:
    - A list of idioms is loaded from a text file. These idioms are later used to identify and handle specific linguistic patterns that might indicate sarcasm.

Feature Engineering

1. Sentiment Analysis:
    - Sentiment scores are calculated for each word in the headline using SenticNet and SentiStrength libraries. These scores help in understanding the positive and negative sentiments conveyed by the words.
2. Contradiction Detection:
    - The presence of contradictory sentiments within a single headline is identified. This is based on the co-occurrence of positive and negative sentiment scores.
    - Coherence checks are performed to see if different parts of a sentence refer to the same subjects or entities, which can help in detecting subtle sarcasm.
3. Linguistic Features:

- Various linguistic features such as the number of emoticons, repetitive punctuations, sequences of capital letters, boosters, slangs, and idioms are extracted from the headlines.
- These features are used to create a detailed feature set that captures different aspects of the text that might indicate sarcasm.

Model Training

Model Training

Support Vector Machine (SVM)

Two SVM models are trained:

1. **SVM on Lemmatized Headline DataFrame (n-gram only)**:
    - This model uses the n-gram representation of the lemmatized headlines for training.
2. **SVM on Feature Set DataFrame (feature set only)**:
    - This model utilizes the detailed feature set created from various linguistic features, including contradiction and coherence.

Naive Bayes

- A Naive Bayes model is also trained using the n-gram representation of the headlines.

Additional Models

1. **Feature Set (only including Contradiction and Coherence)**:
    - This model focuses on the contradiction and coherence features derived from the headline data.
2. **N-gram + Feature Set**:
    - This model combines both the n-gram representation and the detailed feature set for training.
3. **Uni-gram**:
    - This model uses uni-gram representation of the headlines.
4. **Uni-gram + Contradiction**:

    - This model combines the uni-gram representation with contradiction features.

Results and Evaluation

1. Prediction:
    - The SVM models are used to predict the sarcasm labels on the test data.
    - Predictions from both models are compared to understand their performance.
2. Metrics Calculation:
    - Standard metrics such as recall, precision, F1-score, and accuracy are calculated to evaluate the models.
    - Special attention is given to contradiction metrics to see how well the model detects sarcastic contradictions.

Models	**Accuracy**	**Recall**	**Precision**	**F1 score**
Feature Set (only including Contradiction and Coherence)	55.25%	56.15%	49.15%	52.42%
SVM on Lemmatized Headline Dataframe(n-gram only)	85.10%	81.70%	83.93%	82.80%
SVM on Feature Set DataFrame(feature set only)	53.85%	59.31%	59.83%	56.45%
N-gram + Feature Set	83.25%	76.02%	84.28%	79.94%
Uni-gram	82.32%	78.43%	80.75%	79.57%
Uni-gram + Contradiction	69.00%	68.50%	68.50%	68.50%
Naive Bayes	85.00%	84.50	84.50%	84.50%

Integration and Deployment

1. Model Saving:
    - The trained models are saved using joblib for later use and deployment.

2. Streamlit Deployment:
    - The project is intended to be deployed using Streamlit, an interactive web application framework, allowing users to input headlines and get sarcasm predictions in real-time.

Limitations

- Dependency on External APIs and Tools:
    - ConceptNet API: The conceptNet function relies on an external API, which can lead to issues such as network dependency, rate limiting, and potential changes in the API.
    - SenticNet and SentiStrength: These tools require local installation and configuration, which can introduce compatibility issues and affect reproducibility.
- Data Handling and Preprocessing:
    - Manual Feature Engineering: The process of creating features like sentiment scores, punctuation counts, and idiom detection is manually intensive and can introduce biases. This approach also lacks scalability and adaptability to new datasets or changes in data distribution.
    - Lemmatization: While lemmatization reduces inflectional forms, it may oversimplify text, potentially losing nuance in sarcasm detection.
- Feature Set Limitations:
    - Fixed Idioms and Slangs List: The code relies on predefined lists of idioms and slang words. This approach may not capture new or evolving language trends, limiting the model's ability to generalize.
    - Basic Sentiment Analysis: Sentiment scores are calculated using simple heuristics, which may not capture the complexity of sentiment in sarcastic text. Advanced sentiment analysis models could improve this aspect.
- Efficiency and Scalability:
    - Computation Time: The code mentions that the entire process could take 15-20 hours, indicating inefficiencies. Optimizing the code for performance, parallel processing,

  or using more efficient libraries could address this limitation.
- Large Data Handling: The approach of splitting the data into smaller chunks for processing suggests challenges with handling large datasets. More efficient data handling and processing techniques are needed for scalability.

Summary

The project leverages advanced NLP techniques and machine learning to create a robust sarcasm detection system. By combining sentiment analysis, contradiction detection, and linguistic feature extraction, the model aims to accurately identify sarcastic headlines, providing valuable insights for applications in content moderation, social media analysis, and more.