Sarcasm & Irony Detection

Team Members: Aditya Raghuvanshi, Satya Swaroop Gudipudi, Rohan Chowdary **Team Number**: 15

Introduction:

In the realm of social media forums and chats, the pervasive use of figurative language (FL) like sarcasm, irony, and metaphor presents a significant challenge to sentiment analysis. The inherent contradictory and metaphorical nuances in these FL expressions complicate their identification in short texts. Particularly, sarcasm, defined as a figure of speech where the literal meaning is replaced by a figurative one – often opposite to the original – stands as a notable challenge in natural language processing. The task at hand is to effectively identify and interpret these FL forms

Literature Review:

Advances in Sarcasm Detection: Recent developments in sarcasm detection are driven by pretrained models, specialized methods, and effective neural architectures, resulting in more accurate and reliable systems.

Detection Techniques: Sarcasm is detected using a variety of methods, each with its own advantages and disadvantages, such as rule-based, machine learning-based, and deep learning-based approaches.

Applications: By filtering or reacting to sarcastic inputs, sarcasm detection finds use in sentiment analysis, social media analysis, dialogue systems, and human-computer interaction.

Neural Approaches: Neural network approaches, such as two-stage extractive-abstractive models and end-to-end architectures, have become more popular in sarcasm detection, enabling improved representation learning.

This literature review emphasizes key facets of sarcasm detection, emphasizing current developments, methodology, applications, and the function of neural approaches in this area.

Methodology:

Baseline Model:
We will start with baseline models by implementing sequence to sequence models like BiLSTM with attention or with transformer models. Since we are handling sarcasm detection which is a classification based we will train encoder transformer based models and evaluate the performance.

Model Enhancements:
Explore certain latest techniques to improvise the baseline model performance as below:
Fine-Tuning: Given the dataset constraints, we will start replacing our baseline with pretrained embeddings, such as BERT, and continue training them on our sarcasm prediction dataset to adapt the model to the specifics of this task.
Transfer-Learning: As a part of our transfer learning strategy, we will leverage models that have been previously trained on related tasks, like sentiment analysis, from sources such as the Mteb leaderboard. We will then further fine-tune these models to adapt to our specific sarcasm detection task. [6]
Transformer Architecture: Although for our task encoder architecture is best suited, we will explore other architectures like encoder-decoder based model architectures like T5 etc as such model architectures could capture more nuanced representations of the underlying data.

Ensemble Modelling: Experiment with different transformer architectures like Bert, XLNet, GPT etc and perform either mean voting or rank average ensemble where individual model scores are ranked and then the average rank is taken as the final prediction.
Contrastive Learning: For novelty, we will explore and implement SetFit[5] for contrastive learning. By doing this, the model might learn better representations by distinguishing between various classes. This approach is particularly useful to handle severe imbalance in the dataset.

Novelty: Along with above model enhancements we will be approaching code-mix sarcasm detection especially for Hinglish data where the sarcasm tweet is often mixed with both languages. And this is one of the challenge faced by digital platforms. We will try various approaches either by converting Hinglish to English as a translation problem or by training a static embeddings specific to Hinglish text[3].

Evaluation: Evaluate the overall effectiveness of our approach by calculating metrics like F1, Precision, Recall and also loss functions like cross entropy loss.

Challenges:

Following are some challenges with developing sarcasm detection models

Ambiguity and Nuance: Sarcasm often relies on subtleties in tone and context that can be hard to capture with models trained purely on text data. The very nature of sarcasm is to convey the opposite of the literal meaning, which can be challenging for models to understand without context.
Dependency on Context: Sarcasm can be context-dependent. The same sentence might be sarcastic in one context and genuine in another. This makes it hard to have a one-size-fits-all model.
Lack of Labeled Data: There's a shortage of labeled datasets specifically for sarcasm detection. Building a large, high-quality annotated dataset is time-consuming and requires human reviewers to understand nuanced sarcasm.
Cultural and Linguistic Variations: Sarcasm can vary across different languages and cultures. What's considered sarcastic in one culture might not be in another. Similarly, direct translations of sarcastic sentences might lose their sarcastic intent in the process.
Multiple Forms of Figurative Language: Sarcasm is just one form of figurative language. Irony, metaphors, similes, and other figures of speech can further complicate the detection process.

Datasets:

We will be leveraging following datasets for training and evaluation.

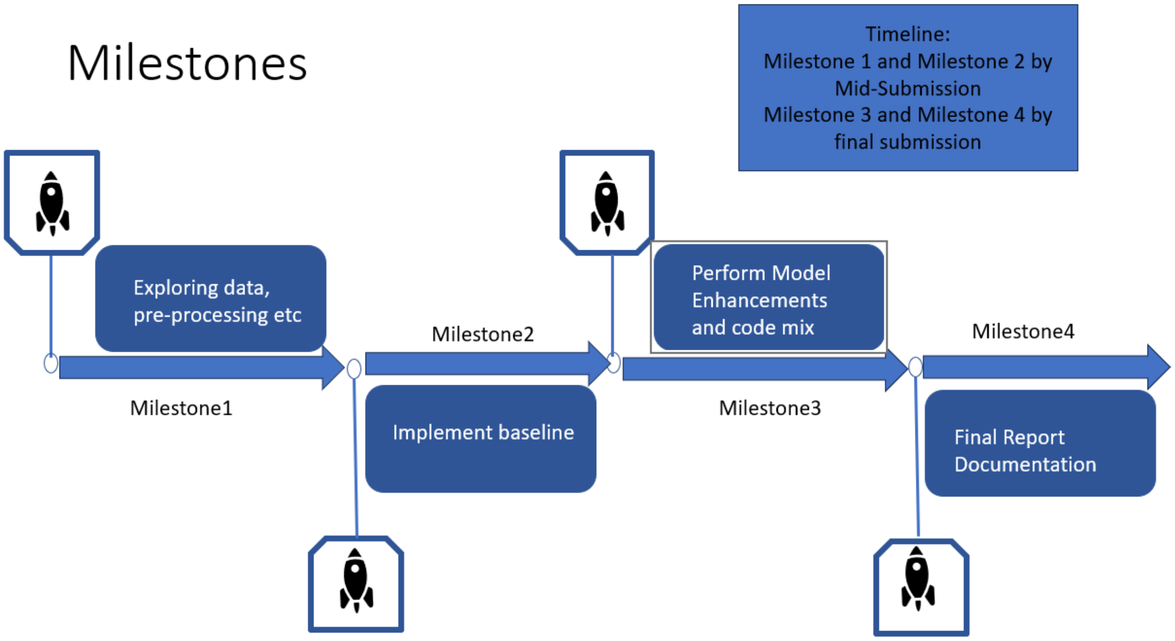
The major challenge with sarcasm detection is availability of data as highlighted in challenges. We have listed some of the datasets and especially benchmark results are available for SemEval-2018 Task3 dataset and this dataset is having 3834 tweets for training and 784 tweets for testing.
The Reddit dataset has approximately 1.3 million samples. The quality of the data to be assessed at model development time.

- SemEval-2018 Task3 datasets
- The Shared Task (2nd FigLang Workshop at ACL 2020)
- Sarcasm on reddit
- News Headlines dataset for Sarcasm Detection

- Code-Mix sarcasm detection

Timelines:

Milestone 1: Explore data and apply data preprocessing
Milestone 2: Implement baseline models by training BiLSTMs and transformer models on the given datasets. (**2nd Oct, 2023**)
Milestone 3: Perform the listed model enhancements and record the individual evaluations along with hyper parameter tuning. Also Extend the implementation to handle code mix datasets.
Milestone 4: Document the final report with findings and observations. (**2nd Nov, 2023**)


Milestones
Timeline:
Milestone 1 and Milestone 2 by Mid-Submission
Milestone 3 and Milestone 4 by final submission
Exploring data, pre-processing etc
Milestone1
Milestone2
Implement baseline
Perform Model Enhancements and code mix
Milestone3
Milestone4
Final Report Documentation

Conclusion:

The approach aims to start with a baseline and then methodically enhance its capabilities through advanced techniques. We will compare our implementation results with benchmark model performance that is based on RCNN-RoBERTa[1]. Our strategy emphasizes the importance of acquiring accurate and meaningful word representations. This not only addresses the inherent challenges posed by irony and sarcasm detection but also ensures that our model captures nuances in context and sentiment. And by comparing with human judgments, we aim to make the model's predictions more aligned with human perceptions of sarcasm.

References:

1. A transformer-based approach to irony and sarcasm detection
2. Sarcastic or Not: Word Embeddings to Predict the Literal or Sarcastic Meaning of Words
3. How Effective is Incongruity? Implications for Code-mix Sarcasm Detection
4. Efficient Estimation of Word Representation in vector space
5. Efficient Few-Shot Learning Without Prompts
6. MTEB: Massive Text Embedding Benchmark: Niklas Muennighoff, Nouamane Tazi, Loïc
    i. Magne, Nils Reimers [For dataset and embedding reference]