MACHINE LEARNING ENGINEERING CAPSTONE PROJECT

Anah Veronica Immanuel

June 3rd 2020

Sarcasm Detection using NLP (Kaggle)

Domain Background: Sarcasm detection is often considered as the Achilles heel of sentiment analysis. **Sarcasm** is “a sharp, bitter, or cutting expression or remark”. The use of irony to mock or convey contempt. The speaker deliberately states the opposite of what she conveys to mock or convey contempt. This can be hard for a traditional sentiment analysis model which assumes that the speaker speaks implicitly and that there is no underlying meaning or context to the message. There are ways however to detect sarcasm based on their common characters

- Co-existence of positive and negative
- Incongruity: Number of times a positive word is followed by a negative word
- Pragmatic features: Analysis of unigrams, bigrams or n-grams for polarity change.
- Pragmatic features: Inclusion of emoticons, excessive punctuation (e.g.!!!!!!) and capitalisation

These features in text can give us some clue about whether a comment/text is sarcasm or not.

Problem Statement: To build a sentiment analysis model specifically capable of identifying sarcasm based on the incongruous property of the text provided.

Dataset and Input: The dataset is provided by Kaggle website and can be accessed using this link: https://www.kaggle.com/rmisra/news-headlines-dataset-for-sarcasm-detection. The dataset consists of news headlines that are labelled as either sarcastic or non-sarcastic. The data set has three attributes:

- **is_sarcastic:** 1 if the record is sarcastic otherwise 0
- **headline:** the headline of the news article
- **article_link:** link to the original news article. Useful in collecting supplementary data

Solution Statement: First step of the solution is to load and explore the data. Based on insights, I will then preprocess the data by tokenization, stop words removal, lemmatization, Part of speech (POS) tagging, NER (Named entity recognition tagging) and word embeddings. Features engineering to include incongruence measure, overall polarity of the statement will also be included. The data can now be split into train and test sets. The model will then be trained using a binary classifier to distinguish sarcasm and non-sarcasm text based on its features. I will try out binary classifiers like logistic regression, decision trees, random forest, SVC and choose deep learning models like CNN if required based on the model's performance.

Benchmark Model: The benchmark model for this solution will be logistic regression. In this dataset, the occurrence of sarcasm and non-sarcastic headlines are balanced, so I will try to beat the probability of its occurrence.

Evaluation Metric: Since the dataset is balanced, accuracy of the predicted values vs the true values make more sense. However, since sarcasm is not common in day to day life, metrics such as recall, precision and f1 score can also be used to evaluate how the model might generalize.

Project Design: The raw data will first be loaded in, explored and analyzed. The data will be preprocessed (tokenization, stop words, lemmatization, POS, NER, and word embeddings). Feature engineering to introduce new features will be done if the data requires it. The data will then be split into train and test sets. The model will then be trained on binary machine learning classifiers and deep learning algorithms if deemed necessary. The model that performs best on the test set will be the final model for submission.

References:

- https://www.researchgate.net/publication/325843750_A_COMPREHENSIVE_STUDY_ON_SARCASM_DETECTION_TECHNIQUES_IN_SENTIMENT_ANALYSIS
- https://www.slideshare.net/anujgupta5095/sarcasm-detection-achilles-heel-of-sentiment-analysis
- https://www.kaggle.com/rmisra/news-headlines-dataset-for-sarcasm-detection