**CSCE 5290 : Natural Language Processing**

**Project Proposal**

**Title : Twitter Emotion Classification**

**Github link :** <https://github.com/vijayrekala/5290-Project>

**1. Motivation :**

The increasing usage of social media platforms such as Twitter has resulted in a massive collection of data that reflects human emotions and feelings in real time. Understanding these emotions has several applications including marketing techniques and mental health support systems. However, manually evaluating such large amounts of data is impractical. As a result, using Natural Language Processing (NLP) techniques to automate emotion classification on Twitter data will greatly improve our ability to extract valuable information from social media.

**2. Significance :**

This project is significant because it addresses the demand for automated sentiment analysis in a dynamic and real-time environment like Twitter. The capacity to accurately define emotions in tweets can be useful in a variety of industries, including market research, political analysis and mental health monitoring. Furthermore, understanding the emotional pulse of online communities may help to build specific interventions and personalized services.

**3. Objectives :**

* Creating a powerful NLP model that can properly classify tweets into six different emotions: anger, fear, joy, love, sadness, and surprise.
* Extensive model training and optimization are required to achieve high levels of accuracy, precision, and recall in emotion classification.
* Create a user-friendly model that make the predictions on the tweets based on the emotions used within the tweet message.
* Evaluating the model's performance using relevant metrics and prove its effectiveness with regression testing on Twitter datasets.

**4. Features :**

* Emotion categorization is carried out using cutting-edge NLP approaches such as word embeddings, Recurrent Neural Networks (RNNs) and attention mechanisms.
* Creating a scalable and efficient data preprocessing pipeline to handle massive amounts of Twitter data including text cleaning, tokenization and feature extraction.
* Implementing transfer learning methodologies to improve performance by leveraging pre-trained language models such as BERT.
* Integrating interpretable methodologies to provide insights into the models decision-making process which improves transparency and trustworthiness.
* The model is regularly monitored and updated to reflect changing language patterns and user behaviors on Twitter.
* The milestone defined for this project is 4 weeks, by providing each week for pre-processing the data, RNN’s, defining six emotion classes, evaluation metrics.

**5. Dataset :**

The dataset for this project consists tweets tagged with six different emotions: anger, fear, joy, love, sadness and surprise. The dataset is sourced from Kaggle and includes a broad mix of tweets from various people. It's a csv file of 733kB. It combines textual material with matching emotion labels, allowing for supervised emotion classification. Preprocessing activities such as text normalization, special character removal and tokenization will be conducted before model training to ensure that the dataset is suitable for analysis.

**6. Visualization :**

**6.1 Data Pre-Processing :**

1. Get the Twitter dataset that comprises tweets labeled with the following six emotions: fear, rage, joy, sadness, love and surprise.
2. To make sure the text data is consistent and clean, do text cleaning activities such eliminating punctuation, special characters, URLs and non-alphanumeric letters. Tokenize the cleaned text data dividing it into discrete words or tokens so that it can be processed further.
3. To reduce noise and increase the effectiveness of subsequent NLP tasks, remove common stopwords from the tokenized text (such as "the" ,"is" ,"and").
4. To standardize the text representation, handle contractions and abbreviations remove accents and convert all letters to lowercase.
5. To represent the text numerically for machine learning algorithms, extract specific characteristics from the preprocessed text data such as word embeddings, TF-IDF (Term Frequency-Inverse Document Frequency) vectors or n-gram representations.
6. The preprocessed dataset need to be divided into training and testing.

**Data Pre-Processing Workflow**

Data Collection

(Dataset)

Text Cleaning

Tokenization

Stopword Removal

Split data to

training & testing

Feature

Extraction

Text

Normalization

**6.2 Model Architecture :**

1. *Input layer*: We take the pre-processed data as input.
2. *Word Embedding Layer*: This layer captures the semantic links between words and phrases by embedding the input text data into dense vector representations.
3. *Recurrent Neural Network (RNN)*: This type of neural network processes sequential input by using recurrent connections to identify temporal dependencies in the text.
4. *Emotion Classes*: Enables multi-class classification by generating probability distributions over the six emotion classes (anger, fear, joy, love, sadness and surprise).
5. *Loss Function and Optimization*: Backpropagation and gradient descent-based optimization methods are used to calculate the loss between the predicted and actual emotion labels and to optimize the models parameters.
6. *Evaluation Metrics*: Uses a variety of evaluation measures including accuracy, precision, recall and F1-score to assess the models performance.

**Model Architecure Workflow**

Pre-processing Data

Word Embedding

Recurrent Neural

Network (RNN)

Six Emotion Classes

Loss Function &

optimization

Evaluation Metrics

Recall

Accuracy

Precision