Sentiment Analysis Using 1D CNNs

1. Introduction

In this report, we present an approach to sentiment analysis using Convolutional Neural Networks (CNNs) applied to Twitter data. We discuss the preprocessing techniques employed, the word embedding algorithm utilized, the CNN architecture adopted, and the results achieved.

2. Dataset Preprocessing

We begin by preprocessing the Twitter dataset to prepare it for sentiment analysis. The following techniques are applied:

- **Lowercasing**: Convert all text to lowercase to ensure consistency in text representation.
- **URL Removal**: Replace URLs with a generic **<url>** token to remove hyperlinks, as they do not contribute to sentiment analysis.
- **Username Replacement**: Replace usernames (mentions) with **<user>** token to maintain anonymity and remove personalized information.
- **Emoji Handling**: Replace emojis with its corresponding textual representations to standardize emoji occurrences.
- **Special Character Removal**: Remove non-alphanumeric characters and symbols from the text.
- **Tokenization**: Tokenize the text into individual words for further processing.

3. Word Embedding Algorithm

We employ the Word2Vec algorithm to generate word embeddings from preprocessed text data. Word2Vec utilizes neural networks to learn distributed representations of words in a continuous vector space. These embeddings capture semantic relationships between words and are crucial for capturing contextual information in natural language.

4. CNN Architecture

Our CNN architecture consists of the following components:

- **Embedding Layer**: Initialized with pre-trained Word2Vec embeddings, this layer maps each word token to its corresponding embedding vector.
- **Convolutional Layer**: Applies convolutional filters to extract features from the embedded representations of the text. ReLU activation function is used to introduce non-linearity.
- **Global Max Pooling Layer**: Aggregates the most salient features by taking the maximum value across each feature map.
- **Fully Connected Layers**: One dense layer with ReLU activation function followed by dropout regularization to prevent overfitting.
- **Output Layer**: Final layer with softmax activation function for binary classification (positive or negative sentiment). We used softmax because there are two output neurons.

5. Results

After training the CNN model on the preprocessed dataset, we achieved the following results:

- **Training Accuracy**: 81.9% after 5 epochs
- **Test Accuracy**: 81.9%
- **Some random samples from test data**:

1) on the balcony at kensingtons in the sun ---> positive

2) <user> i ll suffer alongside you ---> positive

3) late lunch then off to cheers for bday drinks ---> positive

4) <user> why not ---> negative

5) <user> i just need to know what mine is ---> positive

6) <user> morning happy sunday ---> positive

7) i need to find my disc so i can update my spyware stuff and anti virus stuff ---> negative

8) ughh my dad is on my case ---> negative

9) <user> i m obviously not meant to send this email out as its crashed again and i was so close to the send button ---> negative

10) i am this close to getting that second job i just have to wait a week ---> negative

6. Conclusion

In conclusion, our approach to sentiment analysis using CNNs and Word2Vec embeddings showcases promising results. By leveraging advanced preprocessing techniques, powerful word embedding algorithms, and a carefully designed CNN architecture, we demonstrate the capability to extract sentiment information from Twitter data effectively.