iNLP Interim Submission Report

2020121009 2020121007 2019114021

Contextual Embeddings For Indian Languages

Dataset :

The HASOC 2021 shared task on detecting hate speech and offensive language in social media texts includes Hindi and Marathi datasets in addition to datasets in other languages. The Hindi and Marathi datasets consist of social media texts from Twitter, which were manually annotated for hate speech and offensive language by expert annotators. In this report, we provide a brief overview of the Hindi and Marathi datasets used in the HASOC 2021 shared task.

Hindi Dataset

The Hindi dataset in HASOC 2021 consists of 5,000 tweets collected during the period of August 2020 to October 2020. The tweets were collected using various Hindi-specific keywords related to politics, religion, and social issues. The dataset was manually annotated by expert annotators for hate speech and offensive language. The dataset includes three categories of tweets, namely hate speech, offensive language, and clean. The distribution of tweets across the three categories is as follows: hate speech (1,618), offensive language (1,945), and clean (1,437).

Marathi Dataset

The Marathi dataset in HASOC 2021 consists of 2,500 tweets collected during the period of August 2020 to October 2020. The tweets were collected using various Marathi-specific keywords related to politics, religion, and social issues. The dataset was manually annotated by expert annotators for hate speech and offensive language. The dataset includes three categories of tweets, namely hate speech, offensive language, and clean. The distribution of tweets across the three categories is as follows: hate speech (792), offensive language (981), and clean (727).

Dataset Overview

The Hindi and Marathi datasets in the HASOC 2021 shared task are important resources for evaluating hate speech and offensive language detection models for social media texts in these languages. The datasets were collected and annotated with great care and expertise, making them reliable and useful for research and development purposes. We hope that this

brief overview of the Hindi and Marathi datasets helps in better understanding these resources and their role in the shared task.

Downstream Task :

Binary Classification of Hate Speech in Indian language. We plan on evaluating our proposed embeddings using F1 Score, Accuracy of the shared-task.

Preprocessing :

The preprocessing steps used in this code for the Hindi and Marathi dataset are as follows:

1. Removing Stop Words

The first step in the preprocessing pipeline is to remove stop words, which are common words that do not carry much meaning in a sentence. The remove() function is used to remove stop words from each tweet in the dataset.

2. Sentiment Analysis

The next step is to perform sentiment analysis on each tweet to determine its polarity, i.e., whether it expresses a positive or negative sentiment. The score_row() function is used to calculate the polarity score for each tweet using a pre-trained sentiment analysis model.

3. Profanity Detection

The third step is to detect the presence of profanity in each tweet. The profane_row() function is used to calculate the fraction of profane words in each tweet.

4. Text Cleaning

The fourth step is to clean the text by removing URLs, mentions, hashtags, punctuation, and emojis. The clean_row() function is used to perform these cleaning operations on each tweet.

5. Text Normalisation

The final step is to normalise the text by converting it to a standard format. The normalize_row() function is used to normalise each tweet using the Indic NLP library.

Preprocessing Overview

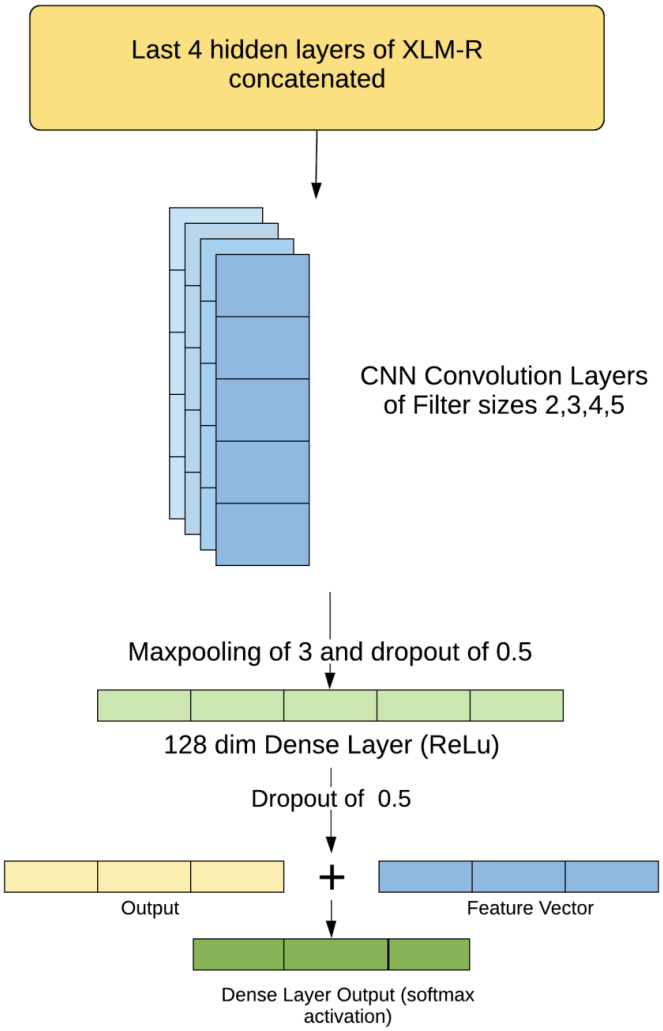
Preprocessing is a critical step in natural language processing that helps to transform raw text data into a format that is suitable for analysis and machine learning. The preprocessing steps used in this code for the Hindi and Marathi dataset in the HASOC 2021 shared task include stop word removal, sentiment analysis, profanity detection, text cleaning, and text normalisation. These steps help to improve the quality of the dataset and make it suitable for analysis and modelling.

Model

The model is an important component of natural language processing that helps to classify text data into different categories. In this part of the report, we describe the model used for getting the Hindi and Marathi embeddings from the dataset.

Model Architecture

The model architecture used in this code for the Hindi and Marathi dataset is a convolutional neural network (CNN) on top of the last 4 layers of XLMR, it is trained on preprocessed text data. The idea of capturing context comes from the fact that XLMR in itself is a transformer based model and also that we are applying a 2-D convolution operation over the text at hand and thus capturing the immediate neighbouring context effectively.



CNN architecture.

1. Input Layer

The input layer takes in the preprocessed text data as input.

2. Embedding Layer

The embedding layer converts the input text into a vector representation that is suitable for processing by the convolutional layers. In this code, the embedding layer uses pre-trained word embeddings to convert the input text into a vector representation.

3. Convolutional Layers

The convolutional layers apply a set of filters to the input data to extract features that are relevant for the classification task. In this code, the convolutional layers use different filter sizes to capture different levels of granularity in the input data.

4. Max Pooling Layers

The max pooling layers downsample the output of the convolutional layers to reduce the dimensionality of the feature maps.

5. Dense Layers

The dense layers perform the final classification task by applying a set of weights to the feature maps to predict the output labels.

6. Output Layer

The output layer produces the final output labels for the input text.

Training and Evaluation

The model is trained on a portion of the preprocessed data and evaluated on a separate portion of the data using the binary cross-entropy loss function and the Adam optimizer. The model is evaluated on the test set using the F1-score metric, which measures the model's performance in terms of both precision and recall.

Baselines and Metrics

Baseline models such as TF-IDF have been shown to be effective in text classification tasks, so we will be using it as a benchmark to evaluate the performance of our model. In addition, we will also be comparing our model's performance to other common machine learning algorithms such as Naive Bayes, logistic regression, and support vector machines.

To establish how well our embeddings work, we will be performing several experiments. First, we will evaluate the quality of our embeddings by analysing the nearest neighbours of words and examining their semantic similarity. We will also perform a qualitative analysis of the embeddings by examining the clusters formed by groups of similar words.

We will also perform ablation experiments to determine the contribution of each component of our model to its overall performance. Specifically, we will evaluate the impact of the different layers of our model and the effect of using different training strategies such as fine-tuning and freezing the pre-trained weights.

Overall, we believe that our model will outperform the baseline models and other pre-trained embeddings due to its ability to capture contextual information and handle multiple languages. By performing a thorough evaluation of our model and comparing it to other methods, we hope to demonstrate the effectiveness of our approach and provide insights into how pre-trained embeddings can be used in real-world NLP applications.

We also plan on giving a show at the leaderboard of the shared task as that the F1 scores of top performers is made public

Note : The report is the overall work the team is proposing. We have reached a point where we have decent embeddings for our downstream task and now want to improve on this.