



TEXT ANALYTICS

2202100



Introduction

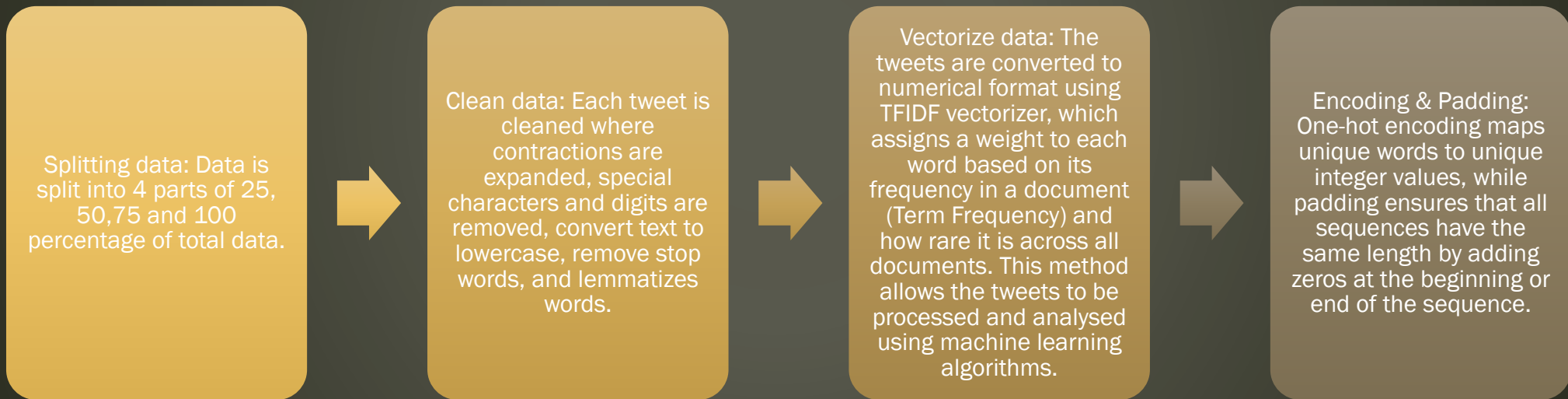
- Problem Statement: The task is to classify Twitter tweets as offensive or non-offensive based on hate speech.
- Approach: Two models, CNN and BLSTM, are used to classify the tweets. The models are evaluated on different data sizes to compare their performance.
- CNN for text classification involves a convolutional layer that applies filters to input text, followed by a max-pooling layer that extracts important features.
- CNNs can automatically learn relevant features from input text, such as n-grams and word embeddings, and handle variable-length input sequences, which is important for text classification tasks.
- BLSTM processes input sequences in both forward and backward directions, allowing it to capture the context of the entire sequence. With memory cells that store information and gates that control information flow, BLSTM can handle variable-length input sequences and capture long-term dependencies between words

Model Selection & justification

CNN: CNN was chosen for its ability to capture local patterns and features in text data, its effectiveness in multi-label classification tasks, and its ability to reduce dimensionality of input data. CNN models can assign one or more labels to a given input, making them well-suited for multi-label classification tasks like hate speech classification.

BLSTM: The BLSTM model was chosen for its ability to capture context, understand the overall meaning of a text sequence, and identify complex instances of hate speech. It can also retain information over longer periods of time, making it useful for hate speech classification. The model's performance on multi-label classification tasks was another key factor, as it allows for accurate categorization of text sequences containing multiple instances of hate speech.

Design & Implementation



Design & Implementation

CNN Model:

Embedding layer with input vocab of 5000 and output embedding vectors of size 32

1D convolutional layer with filters of kernel size 3 and ReLU activation function

MaxPooling1D layer with pooling window of size 2

Dropout layer randomly dropping out 20% of input units

Flatten layer to convert output to a 1D vector

Dense layer with one output unit and sigmoid activation function for binary classification

Trained with Adam optimizer to minimize binary cross-entropy loss

Batch size of 64 and run for 10 epochs

Hyperparameters: vocabulary size of 5000, max tweet length of 100, embedding size of 32, dropout rate of 0.2, Conv1D filters of 32, Conv1D kernel size of 3

Design & Implementation

BLSTM Model:

Embedding layer: input vocab of 5000, output embeddings of size 40, input sequence length of 100

BLSTM layer: 100 units, processes sequence in both directions

Dropout layer: randomly drops out 50% of input units

Dense layer: one output unit with sigmoid activation function for binary classification

Trained to minimize binary cross-entropy loss using Adam optimizer with learning rate of 0.005

Data processed with batch size of 64 and run for 10 epochs

Hyperparameters used: vector features of 40, vocabulary size of 5000, max tweet length of 100, number of LSTM units of 100, dropout rate of 0.5, optimizer of Adam, learning rate of 0.005, epochs of 10, and batch size of 64.

Design & Implementation

- Process flow diagram of CNN and BLSTM model

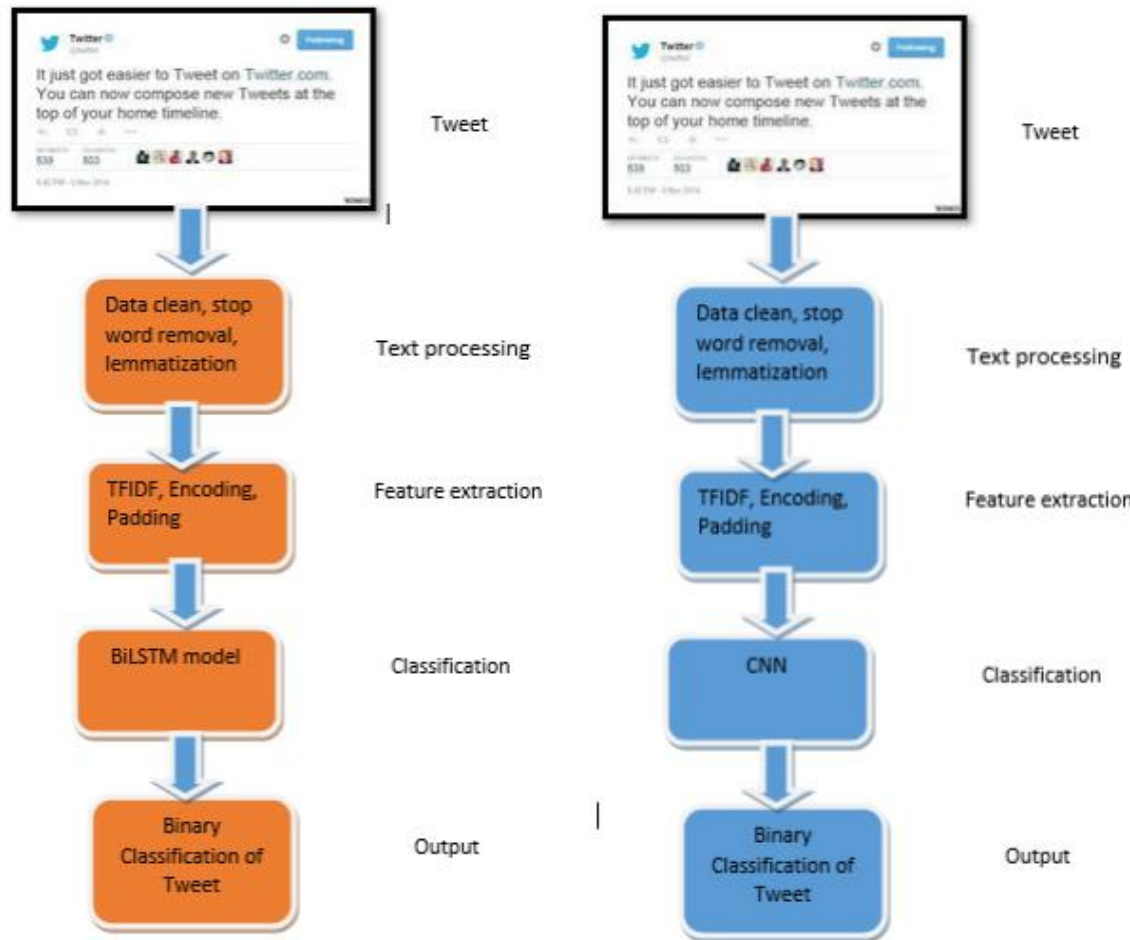


Figure 1: BLSTM Flow diagram

Figure 2: CNN Flow diagram

- CNN and BLSTM stats

Model	F1 Score	Accuracy
CNN	64	71
BLSTM	67	73

Model	F1 Score	Accuracy
CNN	40	66
BLSTM	47	70

- State of the art stats taken from research paper 'Hate Speech Detection Using Convolutional and Bi-Directional Gated Recurrent Unit With Capsule Network'

Compare
model
performance

Model performance comparison

- Model performance

Test Dataset%	CNN	BLSTM
25%	70.3%	65%
50%	71.15%	68%
75%	71.16%	70%
100%	71.17%	73%

- We can observe that the CNN model performed well for 25% and 50% of the data. However, as the size of the data increased, the accuracy of the BLSTM model drastically increased and outperformed the CNN model on 75% and 100% of the data.


Interesting results

Tweet id	Ground Truth	M2 25	M2 50	M2 75	M2 100
15923	OFF	NOT	NOT	NOT	NOT
22452	NOT	OFF	OFF	OFF	OFF
13876	NOT	NOT	NOT	NOT	NOT
83681	OFF	OFF	OFF	OFF	OFF
74791	OFF	OFF	NOT	OFF	OFF

■ CNN:

Tweet id	Ground Truth	M1 25	M1 50	M1 75	M1 100
60133	OFF	NOT	NOT	NOT	NOT
15565	NOT	OFF	OFF	OFF	OFF
13876	NOT	NOT	NOT	NOT	NOT
83681	OFF	OFF	OFF	OFF	OFF
49139	OFF	OFF	NOT	OFF	OFF

■ BLSTM:



Data should be split carefully to incrementally increase the size of the training dataset without losing data integrity or causing duplication.

Lemmatization was used instead of stemming for tokenization to improve results, given that the task involves text classification based on hate speech present in tweets.

Padding was necessary to ensure that the models received the correct size of data batches to run correctly.

Increasing the size of the training data resulted in a higher model performance.

Lessons learned



Conclusion

- Both models performed well in identifying offensive speech
- CNN model performed well for 25% and 50% of the data. However, as the size of the data increased, the accuracy of the BLSTM model drastically increased and outperformed the CNN model on 75% and 100% of the data.
- Factors to consider: dataset size, model complexity, interpretability
- Evaluating models on different data sizes is crucial
- Further exploration and analysis of twitter data may improve model performance and understanding of success factors



THANK YOU