CE807 – Assignment 2 - Final Practical Text Analytics and Report

Student id: 2205233

Abstract

This text describes a 12,313-instance dataset with 33.3% "OFF" and 67.7% "NOT" labeled. Data is split into 4 subsets (25%, 50%, 75%, and 100%) from 12,313 samples. The table helps understand data distribution and picks subset size for training models based on resources performance. It also compares two tweet classification models and evaluates effectiveness. Examples of data with ground truth and Model 2's labels based on different proportions are shown. Model 2 predicts "OFF" for 25% and 100% data, but not 50% in "WhoIsQ." Model 21's performance varies with training data amount.

1 Materials

- Code
- Google Drive Folder containing models and saved outputs
- Presentation

2 Model Selection (Task 1)

For this, I'm using two models that I adapted from the papers Convolutional Neural Networks for Sentence Classification by (Kim, 2014) and Hate Speech Detection Using a Convolution-LSTM Based Deep Neural Network by (Ziqi et al., 2019).

2.1 Summary of two selected models:

(Kim, 2014)With the exception of CNN-rand, baseline models like CNN-static, CNN-non-static, and CNN-multichannel all performed exceptionally well. For the comparison with the baseline model, 14 models were employed. The experimental setup made use of about six different datasets. Pre-trained word vectors were also employed. 100 billion words from Google News served as the training data for these word2vec vectors (Mikolov et al., 2013). Another study was done by (Ziqi et al., 2019) in which they created a simple neural network with an embedding layer, Cnn layer Lstm layer with word vectors like random initialized and with GloVe (Global Vectors for Word Representation) (Pennington et al., 2014).

2.2 Critical discussion and justification of model selection

The CNN architecture was selected due to its ability to proficiently acquire sentence compositional semantics and local patterns. Convolutional Neural Networks (CNNs) are utilized for the identification of local features within images and are highly regarded for their exceptional execution in computer vision applications such as image classification. The writer utilizes an equivalent rationale in the classification of written works, employing Convolutional Neural Networks (CNNs) to apprehend localized patterns in phrases. The Bag-of-Words (BOW) model is frequently adopted as the foundational text categorization method, attributable to its user-friendly nature and widespread usage. The potential of recurrent neural network (RNN) models to identify prolonged dependencies within sentences renders them a subject of significant contemplation. citekim2014convolutional.

The authors elucidate the utilization of a Deep Neural Network (DNN) model as a means to augment the efficacy of hate speech detection by virtue of its inherent capacity to autonomously acquire intricate characteristics from the input textual data. The merging of long short-term memory (LSTM) and convolutional neural networks (CNN) is a logical approach to text analysis as they possess unique strengths that complement each other. While LSTMs are capable of replicating enduring relationships between words in the text, CNNs exhibit superior performance in detecting localized patterns (Ziqi et al., 2019).

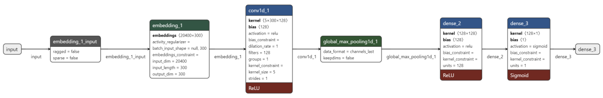

Figure 1: Model-1 Architecture

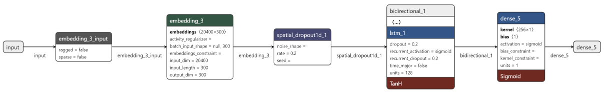

Figure 2: Model-2 Architecture

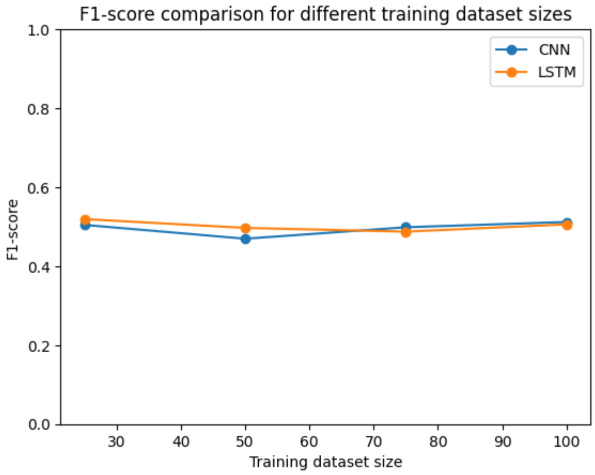

Figure 3: F1-score comparison for different training dataset sizes

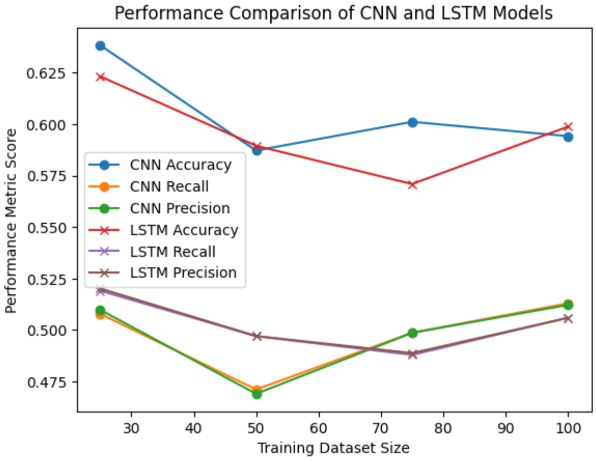

Figure 4: Performance Comparison of CNN and LSTM Models

3 Design and implementation of Classifiers (Task 2)

1. Convolutional Neural Networks for Sentence Classification:

   (a) CNN Network:

   i. Initially, the aforementioned code instantiates a Tokenizer object from the Keras library with the aim of transforming the tweets into sequences of integers. Subsequently, the tokenizer is applied on the training dataset and leverages its word index to generate a lexicon comprising of distinct terms.

   ii. Subsequently, utilizing the Keyed-Vectors class of the Gensim library, the pre-existing word embeddings are retrieved from the compressed binary file GoogleNews-vectors-negative300.bin.gz.

   iii. The present code transforms the tweet sequences into sequences of the uniform length of 300 through padding and converts the labels into arrays constructed with the numpy library.

   iv. The Keras Sequential API is employed to construct the model, which comprises four layers. The initial layer of the model (Figure 1)is composed of an embedding layer that employs pre-existing word embeddings as parameters and is configured to be immutable during training.

   v. The subsequent stratum features a 1-dimensional convolutional layer that incorporates 128 filters, a kernel dimension of 5, and a Rectified Linear Unit (ReLU) activation function. The third stratum is a global maximum pooling layer that retrieves the highest value from each feature map. The fourth layer represents a tightly packed densification with 128 units, coupled with a Rectified Linear Unit (ReLU) activation function. The ultimate stratum consists of a compact output stratum incorporating a sigmoid activation function that facilitates binary classification.

   vi. The constructed model is trained via the implementation of the binary cross-entropy loss function and optimization through the utilization of the Adam optimizer.

vii. The accuracy metric serves as a means of assessing the efficacy of a given model. Subsequently, the model undergoes a training process of two epochs utilizing a batch size of 32. The training and validation datasets are fed into the model through invocation of the fit() function. The chronological account of the model is documented through the utilization of the History callback.

Sr.No	T F1-Score	V F1-Score	Te F1-Score
Model_25%	0.9993	0.7757	0.5131
Model_50%	0.9985	0.7766	0.4466
Model_75%	0.9975	0.7763	0.4964
Model_100%	0.9971	0.7948	0.4828

Table 1: Model-1 performance

Note: T: Train,V:Valid,Te:Test Data

2. Hate speech detection using a convolution-LSTM-based deep neural network:

(a) Pre-processing:

i. Remove the following characters: | : , ; ! ? .

ii. Normalise hashtags into words, hashtags are often used to compose sentences. We use dictionary-based look-up to split such hashtags.

iii. lowercase and stemming, to reduce word inflections.

iv. Removing any tokens with a document frequency less than 5.

(b) The CNN+LSTM network :

i. As CNN is giving the same output in this case, we have only employed the LSTM portion of the algorithm because we have less input that cannot be supplied to it. As a result, we solely used the LSTM component to generate the output.

ii. In the initial phase of the code, pre-existing GloVe word embeddings are extracted from a designated file, after which an embedding matrix is generated. In the matrix, each row corresponds to a distinct word, while the columns denote the various embedding dimensions.

iii. (Figure 2) The formulation of the model entails the utilization of the Keras Sequential API. The initial layer is comprised of an embedding mechanism which leverages the pre-existing word embeddings as the primary weights. The subsequent layer in the neural network architecture consists of a dropout layer in order to avert overfitting, followed by a bidirectional Long Short-Term Memory (LSTM) layer that is capable of assimilating contextual information in both forward and backward directions. The final layer is implemented with a dense output layer that leverages a sigmoid activation function to undertake sentiment evaluation of the input text.

iv. The optimization algorithm employed in the model is Adam. The binary cross-entropy loss function is implemented to enhance performance, whilst the accuracy metric is used to evaluate the successful deployment of the model.

v. The neural network model is conditioned using the training dataset for ten epochs, whereby each epoch consists of dividing the dataset into uniformly sized subsets or batches and utilizing each batch to perform a forward and backward propagation of the data through the network. The batch size utilized in this instance is 128. To monitor the model's progression throughout the training process, callbacks are used to save both the training history and the best model weights.

Sr.No	T F1-Score	V F1-Score	Te F1-Score
Model_25%	0.8241	0.8128	0.5193
Model_50%	0.8558	0.8100	0.4969
Model_75%	0.8698	0.8128	0.4875
Model_100%	0.8680	0.8224	0.5059

Table 2: Model-2 performance

Note: T: Train, V:Valid, Te:Test Data

4 Data Size Effect (Task 3):

1. The "Total" column in Table 3 indicates the total number of samples in each dataset. The "%" columns refer to the percentage of samples that belong to each class: "OFF" (indicating offensive language) and "NOT" (indicat-

ing non-offensive language).The Train dataset contains the highest number of samples, with a total of 12,313 instances. About one-third (33.24%) of the samples in this dataset are labeled as "OFF", while 66.77% are labeled as "NOT".The Valid dataset contains 928 instances, with a similar distribution of "OFF" and "NOT" labels as the Train dataset.The Test dataset contains the lowest number of samples, with a total of 861 instances. However, it has a slightly different distribution of labels compared to the Train and Valid datasets, with 27.99% of samples labeled as "OFF" and 72.00% labeled as "NOT".

Dataset	Total	% OFF	% NOT
Train	12313	33.24	66.77
Valid	928	33.29	66.70
Test	861	27.99	72.00

Table 3: Dataset Details

2. Table 4 shows the distribution of the train data after splitting it into four subsets: 25%, 50%, 75%, and 100%. The total number of samples in the original dataset is 12,313. The table also shows the percentage of tweets that are labelled as "OFF" and "NOT" in each subset. For instance, when the dataset is split into 25%, there are 3,078 samples in the subset. Out of these, 33.23% of tweets are labelled as "OFF", while 66.76% are labelled as "NOT". Similarly, when the dataset is split into 50%, 75%, and 100%, the number of samples in the subsets and the percentage of "OFF" and "NOT" tweets in each subset are shown in the table. This table can be useful for understanding how the distribution of the data changes when using different amounts of training data. It can also help in selecting the appropriate subset size for training a model, depending on the available computing resources and the desired trade-off between model performance and training time.

Dataset	Total	% OFF	% NOT
25%	3078	33.23	66.76
50%	6156	33.23	66.76
75%	9235	33.23	66.76
100%	12313	33.23	66.76

Table 4: Train Dataset Statistics of Different Size

3. Table 5 presents a comparative analysis between two models, namely Model 1 and Model 2, utilizing the complete dataset. The presented table exhibits exemplars alongside their corresponding designations as supplied in the ground truth (GT) document. Each sample instance was categorized by both models, and the outcome of the labelling procedure is delineated in the corresponding column. The table's labels are binary in nature, denoting the classification of a given tweet as offensive (OFF) or non-offensive (NOT). The initial column exhibits the illustrative percentage (Example %), which pertains not to the classification, but rather to the magnitude of the sample employed for the purposes of training and testing the models. The objective of the table is to juxtapose and evaluate the efficacy of two models in accurately categorizing tweets as either offensive or innocuous. This data may be leveraged to appraise the efficacy of the models and subsequently choose the most suitable one for a designated use case.

Example %	GT	M1(100%)	M2(100%)
#WhoIsQ ..	OFF	OFF	NOT
#ConstitutionDay is..	NOT	OFF	NOT
#FOXNews #NRA..	NOT	NOT	NOT
#Watching #Boomer..	NOT	NOT	NOT
#NoPasaran: Unity..	OFF	NOT	NOT

Table 5: Comparing two Model's using 100% data: Sample Examples and model output using Model 1 & 2. GT (Ground Truth) is provided in the test.csv file.

4. In Table 6, the model output of Model 1 is compared across varying data sizes, consisting of 25%, 50%, 75%, and 100% of the full dataset, for selected sample instances. The table additionally comprises the veritable labels (GT) attributed to every instance. The presented table illustrates the impact on the model's performance, in relation to the incremental inclusion of data in the training process. As an illustration, the initial row of data exhibits the ground truth classification as OFF. However, when the model undergoes training with merely 25% of the data, it yields a classification of NOT. Upon complete training, encompassing 100% of the available data, the model accurately classifies the data as OFF. The tabular presentation affords an avenue to

contrast and evaluate the efficacy of the model across varying data sizes, thereby serving as a useful tool for discerning the most favourable dataset magnitude.

5. Table 7 presents a comparative analysis of the efficaciousness of Model 2 by varying the quantity of data available. The presented table delineates diverse exemplary instances associated with their ground truth labels (GT) alongside the projected labels generated by Model 2. The predictions are based on varying proportions of the available data, that is, 25%, 50%, 75%, and 100%. As an illustration, regarding the primary case study in question denoted as "#WhoIsQ", the factual designation is as being OFF, whereupon the computational model estimates OFF as the outcome for both the 25% and 100% sample sizes, yet not for the 50% sample size. The example denoted as "#FOXNews" provides an instance whereby the ground truth label is marked as negative. The model in consideration anticipated a negative outcome for 25% of the data size; however, it predicted a positive outcome for the 50% and 75% data sizes. The present tabular representation facilitates the evaluation of how the performance of Model 2 is influenced by distinct levels of training data allocation.

5 Summary (Task 4)

5.1 Discussion of work carried out:

The provided exposition delineates three tables employed to compare the efficaciousness of two models, Model 1 and Model 2, in the classification of tweets as either offensive or non-offensive.

Table 5 presents instances of exemplars accompanied with their respective ground truth (GT) labels that have been classified by both models, utilizing the entire data pool available. The objective of the current tabular presentation is to conduct a comparative analysis of the efficaciousness of two models and subsequently discern the optimal option tailored to suit a particular application.

Table 6 presents a comparative analysis of the model output generated by Model 1 across a range of data sizes, namely 25%, 50%, 75%, and 100%, based on selected sample instances. The presented table depicts the veritable labels, referred to as ground truth (GT), for each instance and illustrates the variation in the model's performance as an augmented volume of data undergoes incorporation within the process of training. The presented tabular data offers a means to analyze and contrast model performance across various data magnitudes, thus facilitating the identification of the most suitable dataset size.

Table 7 depicts the comparative evaluation of Model 2 in relation to the variant amounts of data utilized. The aforementioned table presents a collection of sample instances, accompanied by their respective ground truth classifications (GT), in addition to the forecasted classifications generated by Model 2, utilizing 25%, 50%, 75%, and 100% of the extant data. The presented tabulated information assists in the assessment of the impact of varying quantities of training data on the effectiveness of Model 2. The data presented in the tables serve as a valuable resource in assessing the efficacy of the two models' abilities to accurately classify tweets as either offensive or non-offensive. The utilization of tables can facilitate the process of discerning the most optimal model and data volume that is suitable for a particular application.

5.2 Lessons Learned:

1. It can be inferred that diverse neural network architectures possess distinct aptitudes that could be harnessed in the realm of text analysis. Convolutional neural networks (CNNs) have demonstrated a superior ability to detect local patterns with high precision. Conversely, recurrent neural networks (RNNs), such as long short-term memory (LSTM) models, possess the ability to capture prolonged dependencies existing between words. By amalgamating these two structures, it is feasible to devise a more efficacious paradigm for the scrutiny of textual content. Moreover, the application of deep neural networks (DNNs) may augment the efficacy of identifying hate speech by enabling the model to independently acquire intricate attributes from the input textual information. The consideration of neural network architectures' strengths is crucial in the selection of an appropriate model for a particular text analysis task.

2. A salient lesson gleaned from this empirical study is the criticality of appraising and juxtaposing diverse models employing variegated amounts of data. The presented tables illus-

Example %	**GT**	**(25%)**	**(50%)**	**(75%)**	**(100%)**
#WhoIsQ..	OFF	NOT	NOT	NOT	OFF
#ConstitutionDay..	NOT	NOT	OFF	NOT	OFF
#FOXNews..	NOT	NOT	NOT	NOT	NOT
#Watching..	NOT	NOT	OFF	OFF	NOT
#NoPasaran..	OFF	OFF	NOT	NOT	NOT

Table 6: Comparing Model Size: Sample Examples and model output using Model 1 with different Data Size

Example %	**GT**	**(25%)**	**(50%)**	**(75%)**	**(100%)**
#WhoIsQ..	OFF	OFF	NOT	OFF	NOT
#ConstitutionDay..	NOT	NOT	NOT	NOT	NOT
#FOXNews..	NOT	OFF	NOT	NOT	NOT
#Watching..	NOT	NOT	OFF	OFF	NOT
#NoPasaran..	OFF	NOT	NOT	NOT	NOT

Table 7: Comparing Model Size: Sample Examples and Model output using Model 2 with different Data Size

trate that the efficacy of the models is subject to noteworthy fluctuations, contingent on the quantity of data employed for training purposes. It is of paramount significance to contemplate the associated compromise between the computational resources that are necessitated for processing copious data quantities and the possible amplification in precision that can be accomplished via augmenting the quantity of data.

3. One valuable insight gained pertains to the significance of assessing models based on a subset of instances possessing established truth labels. The aforementioned facilitates an impartial evaluation of the efficacy demonstrated by diverse models, thus furnishing a fundamental proposition that enables the designation of the optimal model for a given purpose. Assessing models based on a subset of exemplars allows for the identification of possible partialities or inadequacies that require remediation.

4. To summarize, the key insight garnered herein pertains to the criticality of model evaluation and comparison employing variable data portions and a set of instances with discernible, verified ground truth labels in the selection of an optimal model for a given application, and for enhancing its efficacy.

References

Yoon Kim. 2014. Convolutional neural networks for sentence classification.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Z Ziqi, D Robinson, and T Jonathan. 2019. Hate speech detection using a convolution-lstm based deep neural network. *IJCCS*, 11816:2546–2553.