Text Classification: Comparative Analysis of Different Deep Neural Network Architectures

NC STATE UNIVERSITY

By

Cosmo Pernie (ckpernie)

Shashank Shekhar (sshekha4)

Yuvaraj Vivekanandan (yviveka)

Supervised by

Dr. Nagiza Samatova

Table of Contents

1. Introduction

The comparative study of three different deep neural network (DNN) architectures for text classification is analyzed in this project. The following three neural networks are considered:

- CNN: Convolutional Neural Networks
- RNN: Recurrent Neural Networks
- HAN: Hierarchical Attention Networks

We have considered the IMDB 25K reviews dataset for our analysis. Various hyper parameters of each model are fine-tuned and the corresponding final model is built using the hyper parameters that yields the best result.

2. Text Classification

One of the widely used Natural Language Processing & Supervised Machine Learning task in different business problems is **"Text Classification"**, it's an example of Supervised Machine Learning task since a labelled dataset containing text documents and their labels is used for training a classifier.

Some examples of text classification are:

- Understanding audience sentiment from social media
- Detection of spam & non-spam emails
- Auto tagging of customer queries
- Categorization of news articles into predefined topics

An end-to-end text classification pipeline is composed of following components:

Training text: It is the input text through which our supervised learning model is able to learn and predict the required class.

Feature Vector: A feature vector is a vector that contains information describing the characteristics of the input data.

Labels: These are the predefined categories/classes that our model will predict

ML Algorithm: It is the algorithm through which our model is able to deal with text classification (In our case: CNN, RNN, HAN)

Predictive Model: A model which is trained on the historical dataset which can perform label predictions.

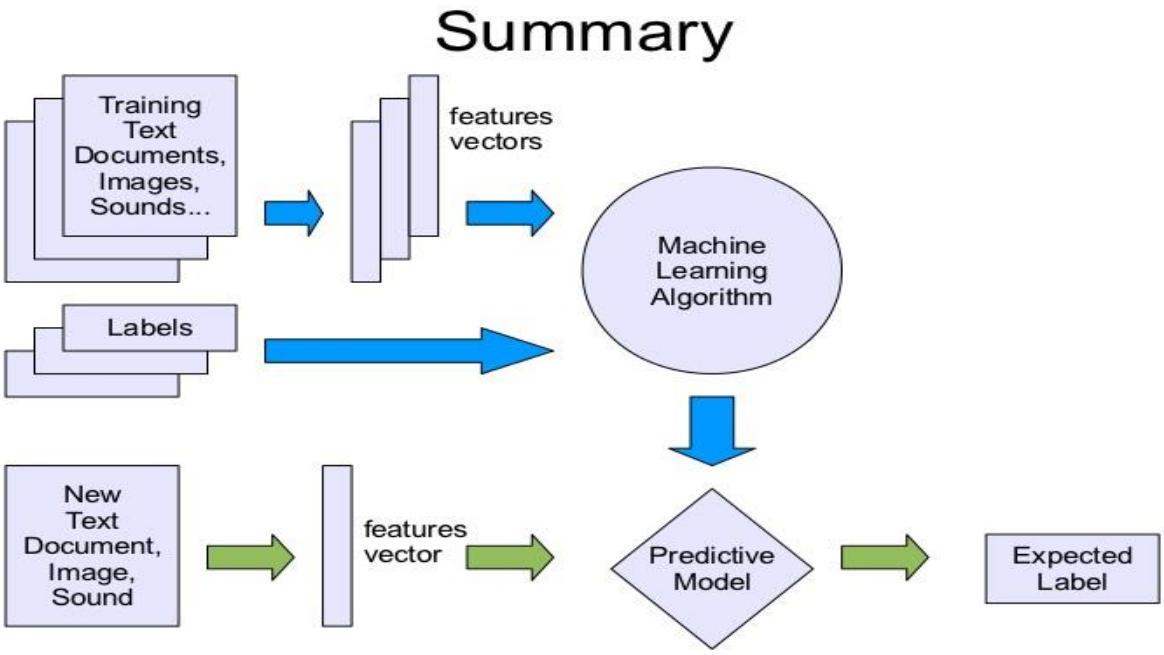


Fig 1: Summary of NLP

3. Dataset Description

Link for downloading the dataset - http://www.cs.cornell.edu/people/pabo/movie-review-data/

The labeled dataset consists of 25,000 IMDB movie reviews, specially selected for sentiment analysis. The sentiment of the reviews is binary, meaning the IMDB rating < 5 results in a sentiment score of 0, and rating >=7 have a sentiment score of 1. No individual movie has more than 30 reviews.

LabeledTrainData - The labeled training set. The file is tab-delimited and has a header row followed by 25,000 rows containing an id, sentiment, and text for each review.

Data fields:

Id - Unique ID of each review

Sentiment - Sentiment of the review; 1 for positive reviews and 0 for negative reviews

Review - Text of the review

Dataset	Training set	Validation set	Testing set
Small	1600	200	200
Medium	8000	1000	1000
Large	20000	2500	2500

Table 1: Summary of dataset

4. Pre-trained GloVe vector

"GloVe is an unsupervised learning algorithm for obtaining vector representations for words (similar to Word2Vec). Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space."

In this project, we have used Glove 6B vector 100d as the pre-trained vector from the below link https://nlp.stanford.edu/projects/glove/

5. CNN Architecture

CNN is a class of deep, feed-forward artificial neural networks (where connections between nodes do not form a cycle) & use a variation of multilayer perceptron's designed to require minimal preprocessing. These are inspired by animal visual cortex. CNNs are generally used in computer vision, however they've recently been applied to various NLP tasks.

Below figure shows the working of CNN on text data through a diagram. The result of each convolution will fire when a special pattern is detected. By varying the size of the kernels and concatenating their outputs, we allow to detect patterns of multiples sizes (2, 3, or 5 adjacent words). Patterns could be expressions like "I hate", "very good" and therefore CNNs can identify them in the sentence regardless of their position.

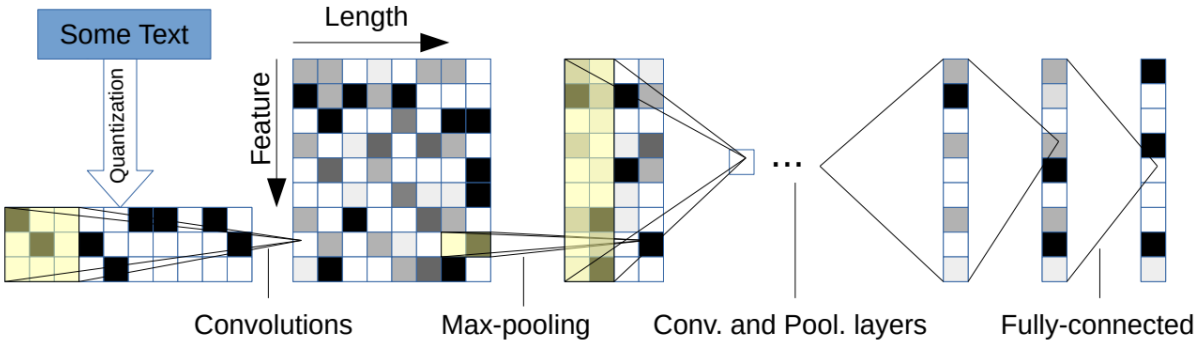
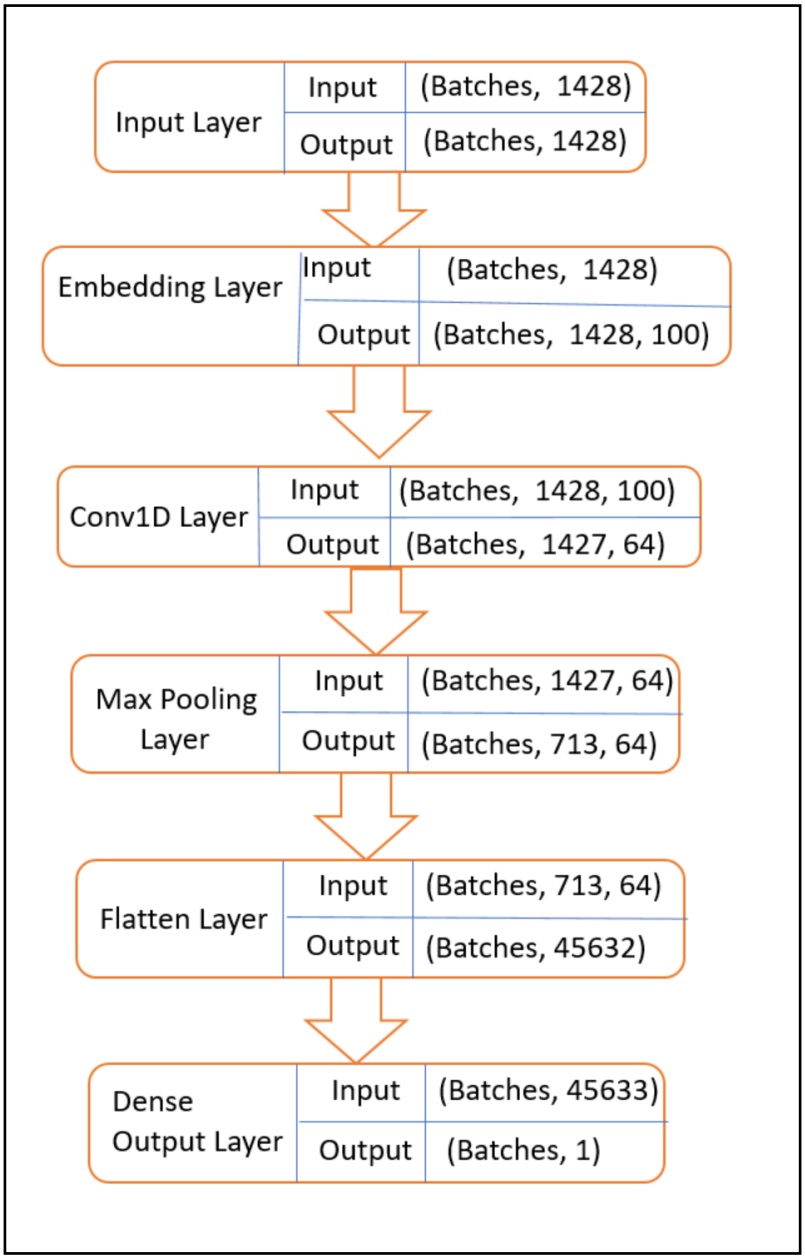


Fig 2: General CNN architecture

Architecture of CNN Model



Hyper-Parameters in CNN model

Hyper-parameter	Values
Activation-Type	Relu, Leaky relu
Pool size	2,3,4
Kernel size	2,3,5
Filter size	64,128,256

Table 2: Hyper-parameters of CNN

6. RNN Architecture

A recurrent neural network (RNN) is a class of artificial neural network where connections between nodes form a directed graph along a sequence. This allows it to exhibit dynamic temporal behavior for a time sequence.

Using the knowledge from an external embedding can enhance the precision of the RNN because it integrates new information (lexical and semantic) about the words, an information that has been trained and distilled on a very large corpus of data. A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor.

By using LSTM encoder, we intent to encode all information of the text in the last output of recurrent neural network before running feed forward network for classification. This is very similar to neural translation machine and sequence to sequence learning.

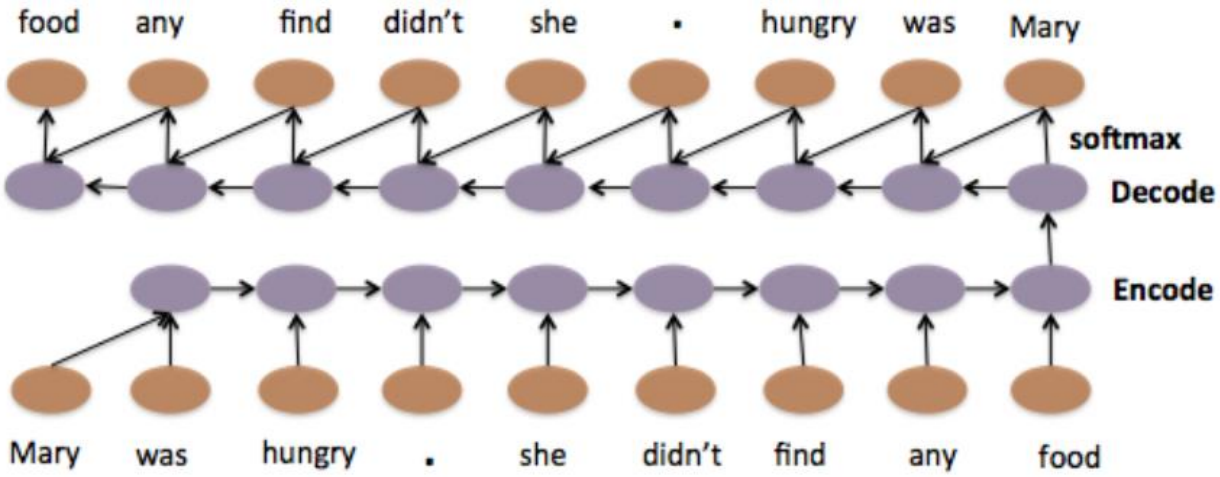
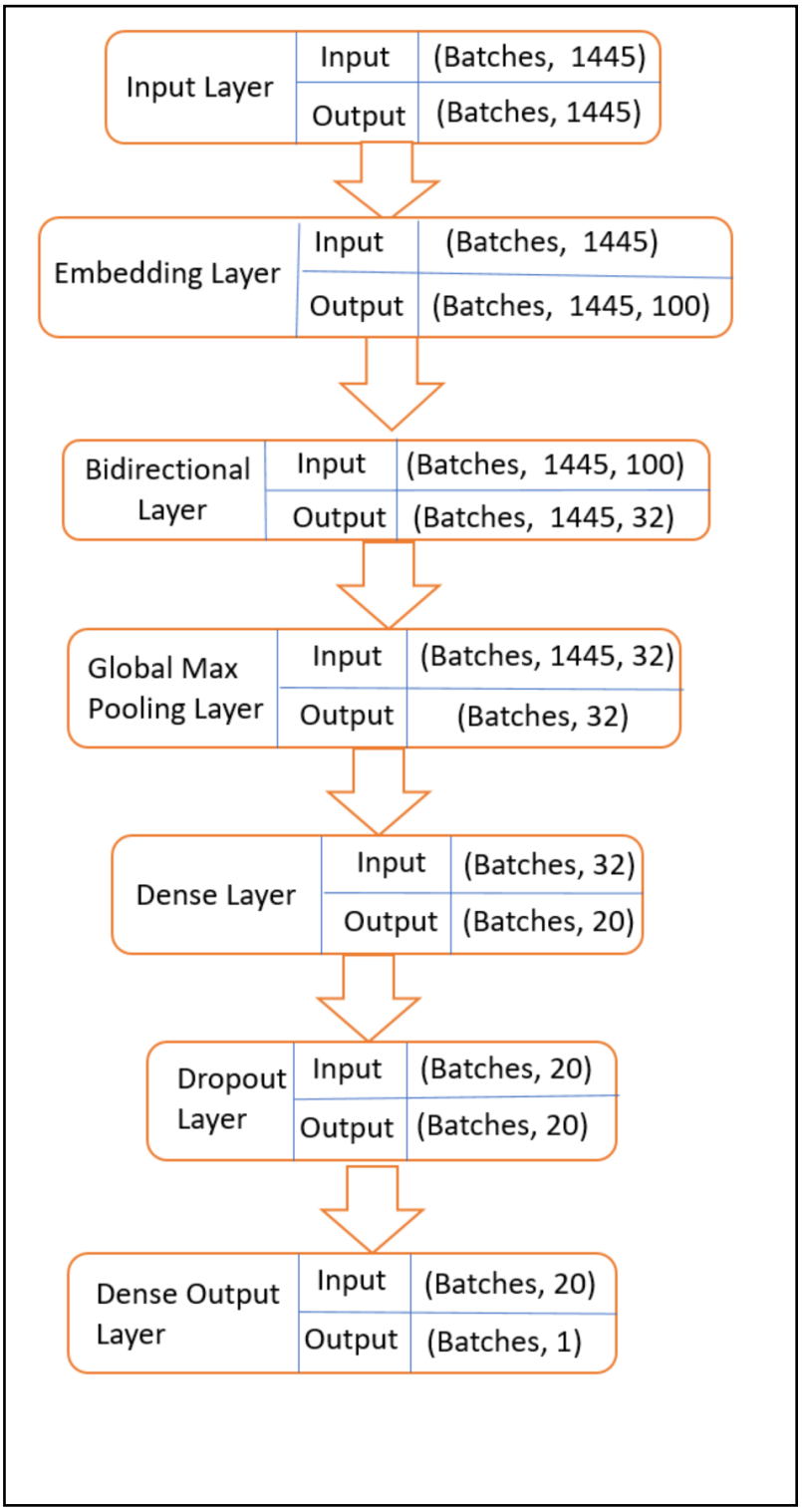


Fig 3: Illustration of RNN model

Architecture of RNN Model



Hyper-Parameters in RNN Model

Hyper-parameter	Values
LSTM units	16,32,64
Dense units	20,40,60
Dropout rate	0.2,0.5,0.8

Table 3: Hyper-parameters of RNN

7. HAN Architecture

The first hierarchical attention networks model was proposed in 2016 and has two distinctive characteristics which set it apart from other models:

- It has a hierarchical structure that mirrors the hierarchical structure of documents
- It has two levels of attention mechanisms applied at the word and sentence level, enabling it to attend differentially to more and less important content when constructing the document representation.

There are 4 main components to the HAN architecture which include

- Word sequence encoder
- Word-level attention layer
- Sentence encoder
- Sentence-level attention layer

For this project the model was constructed using Keras due to its ability to create custom layers for the word and sentence level attention layers.

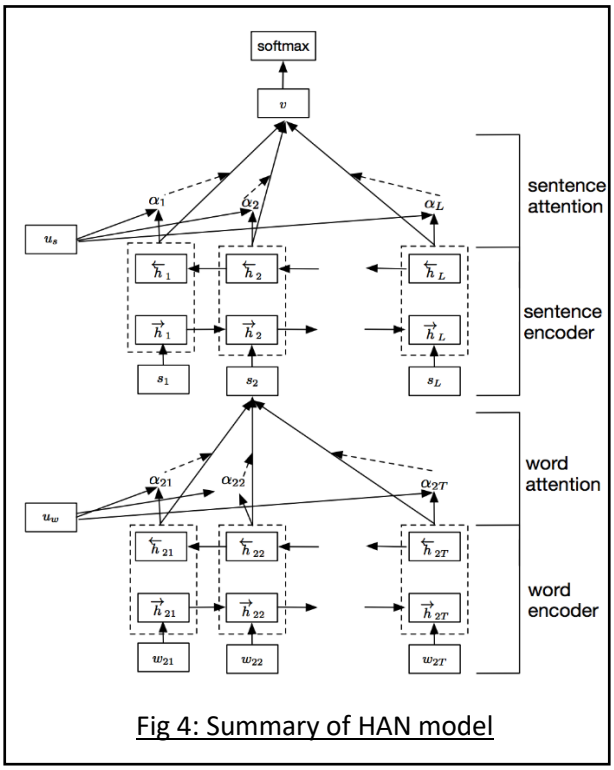
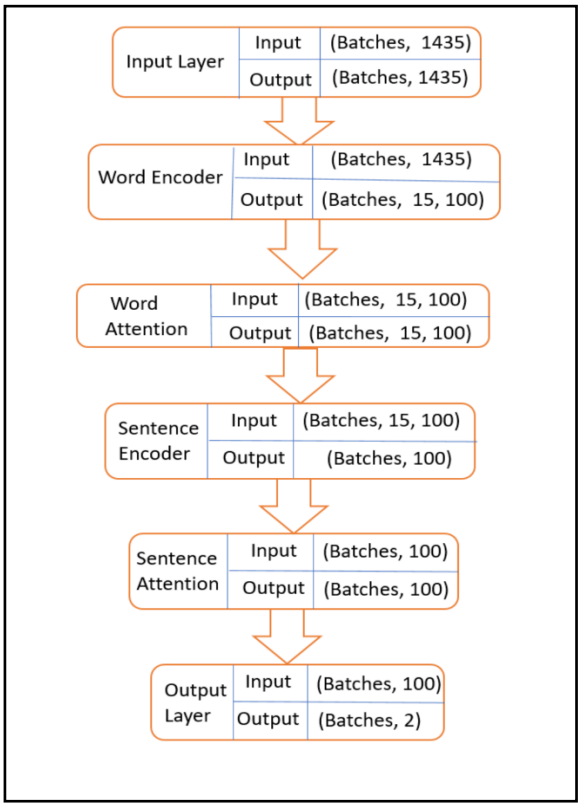


Fig 4: Summary of HAN model

Architecture of HAN Model

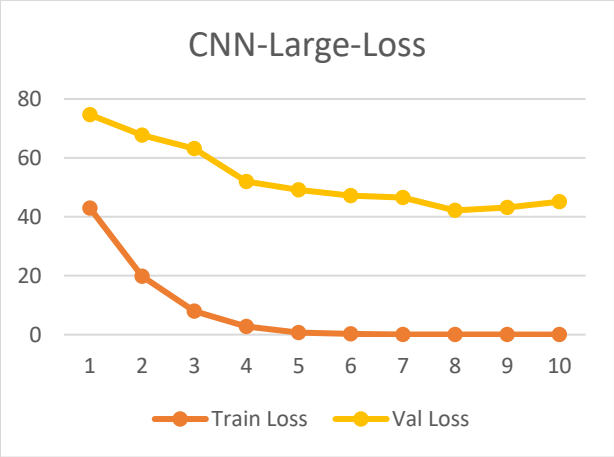


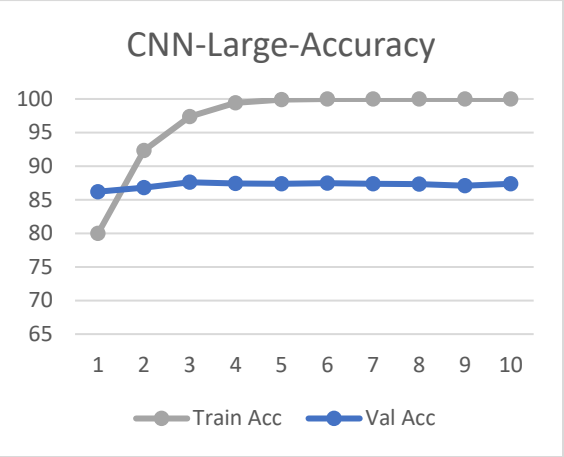
Hyper-Parameters in HAN Model

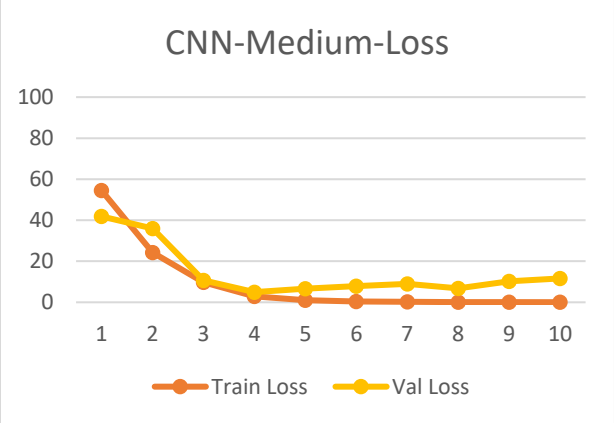
Hyper-parameter	Values
Batch size	32,64,128
Encoding dimensions	100,200,300

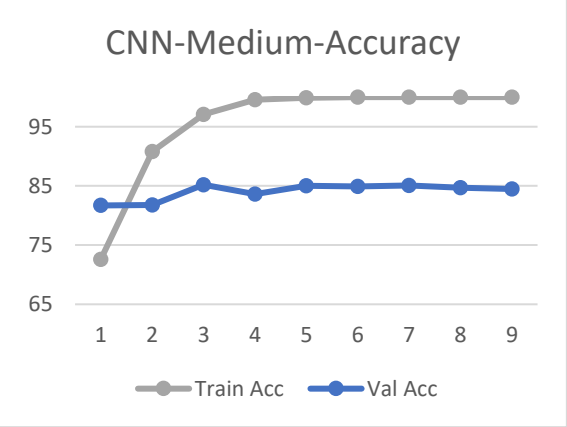
Table 4: Hyper-parameters of HAN

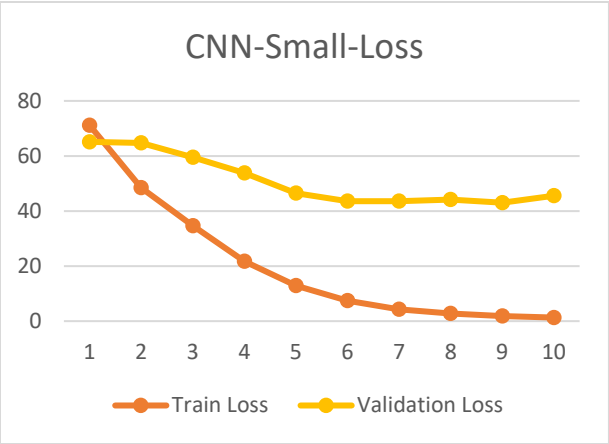
8. Training and Validation Accuracy and Loss over Epochs:

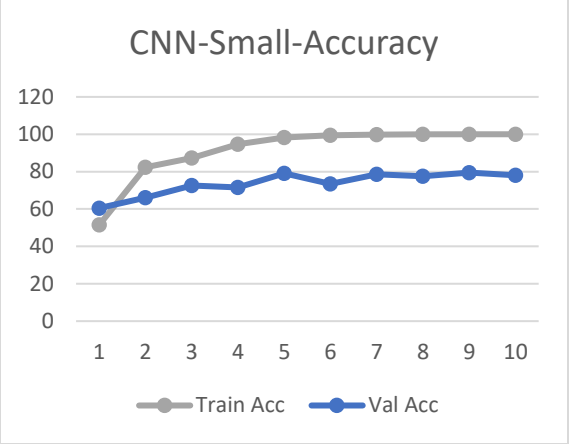


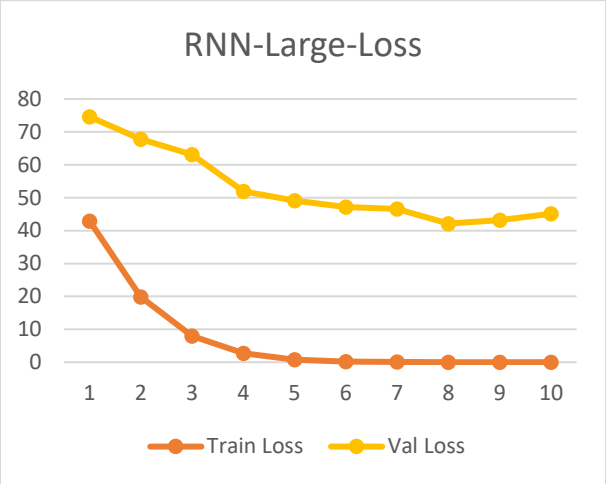


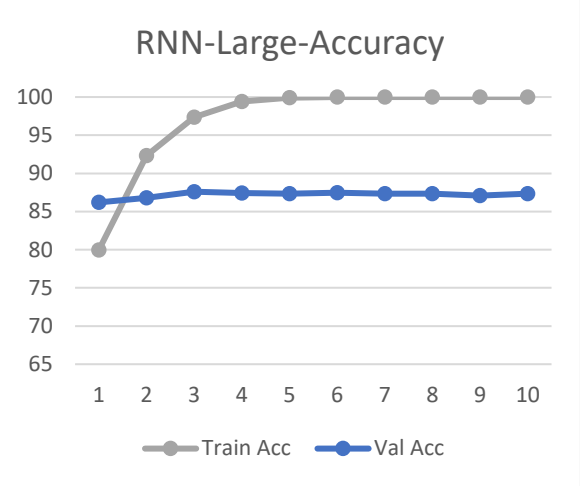


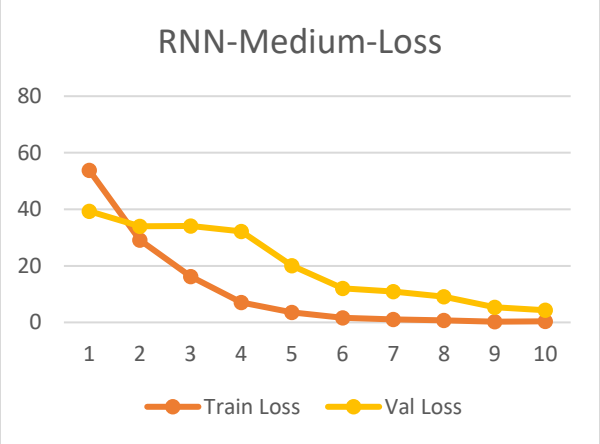


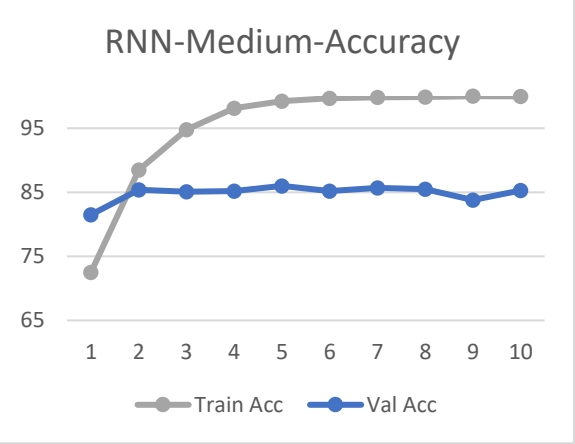

CNN-Small-Loss
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10
Train Loss
Validation Loss

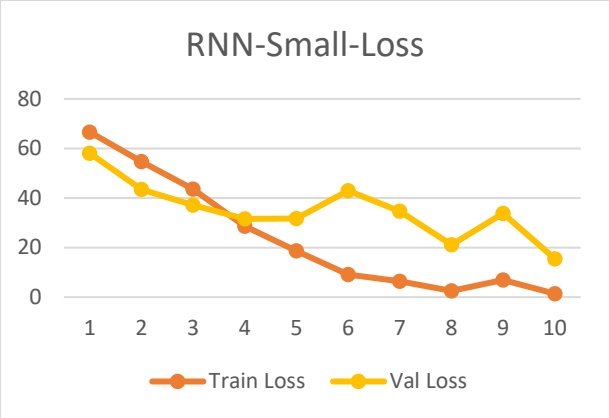

CNN-Small-Accuracy
120
100
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10
Train Acc
Val Acc

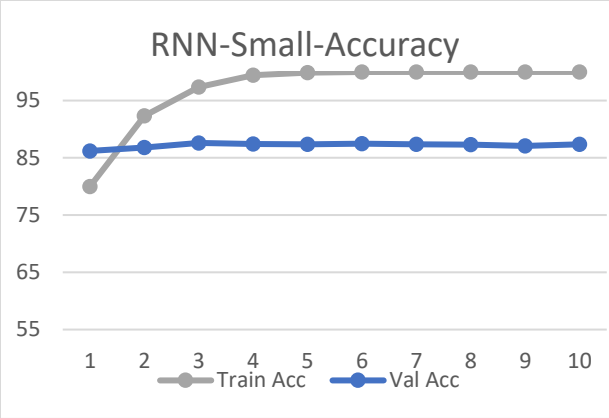

RNN-Large-Loss
80
70
60
50
40
30
20
10
0
1 2 3 4 5 6 7 8 9 10
Train Loss
Val Loss

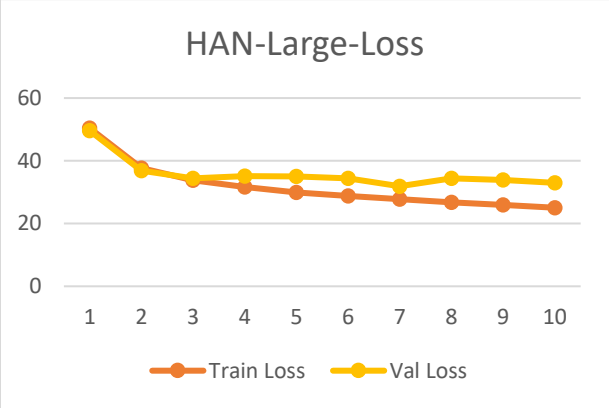

RNN-Large-Accuracy
100
95
90
85
80
75
70
65
1 2 3 4 5 6 7 8 9 10
Train Acc
Val Acc

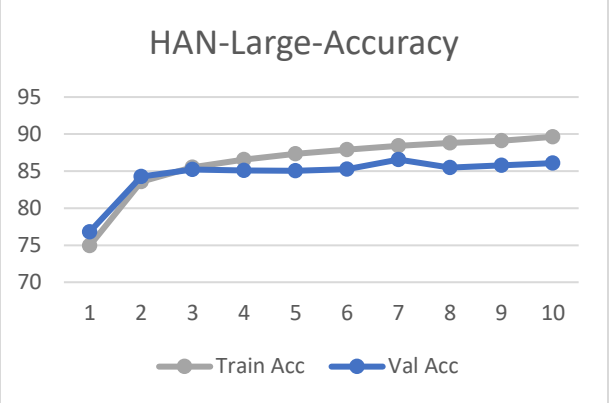

RNN-Medium-Loss
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10
Train Loss
Val Loss

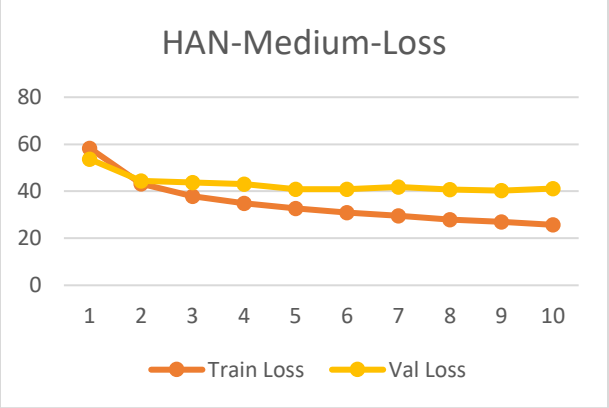

RNN-Medium-Accuracy
95
85
75
65
1 2 3 4 5 6 7 8 9 10
Train Acc
Val Acc

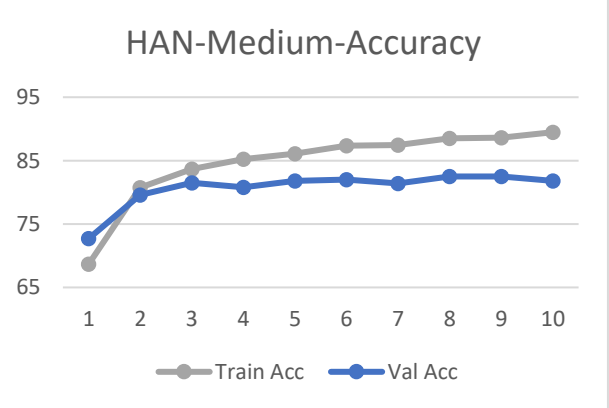

RNN-Small-Loss
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10
Train Loss
Val Loss

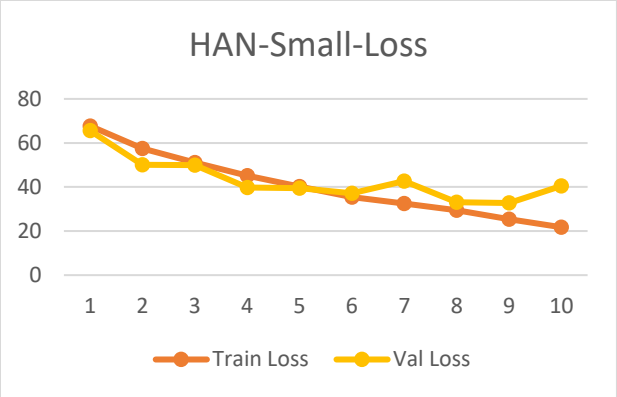

RNN-Small-Accuracy
95
85
75
65
55
1 2 3 4 5 6 7 8 9 10
Train Acc
Val Acc

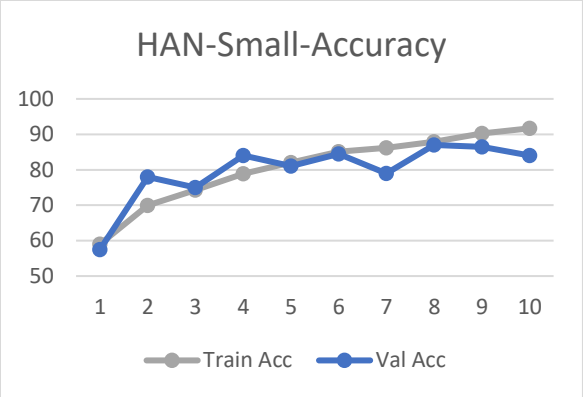

HAN-Large-Loss
60
40
20
0
1 2 3 4 5 6 7 8 9 10
Train Loss
Val Loss


HAN-Large-Accuracy
95
90
85
80
75
70
1 2 3 4 5 6 7 8 9 10
Train Acc
Val Acc

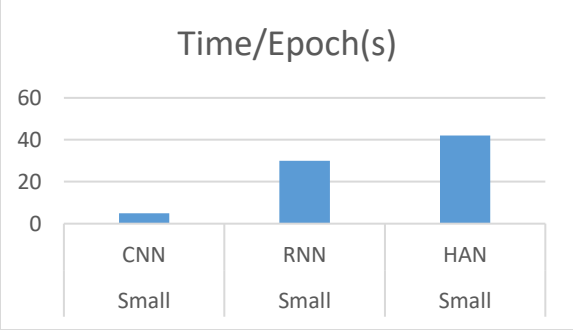

HAN-Medium-Loss
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10
Train Loss
Val Loss

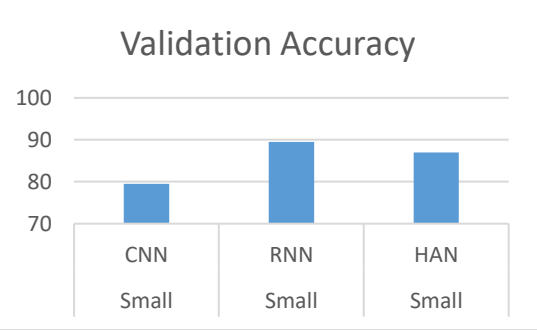

HAN-Medium-Accuracy
95
85
75
65
1 2 3 4 5 6 7 8 9 10
Train Acc
Val Acc

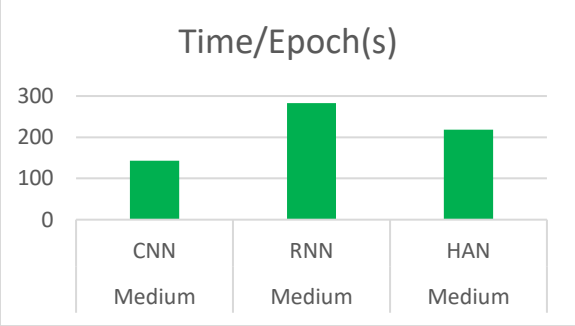


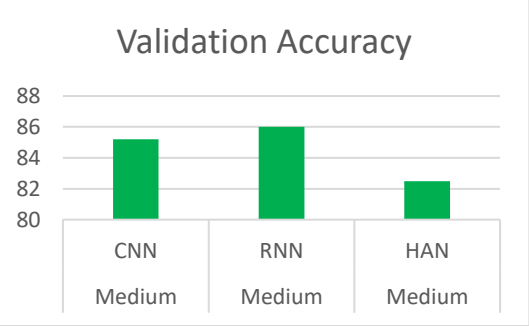


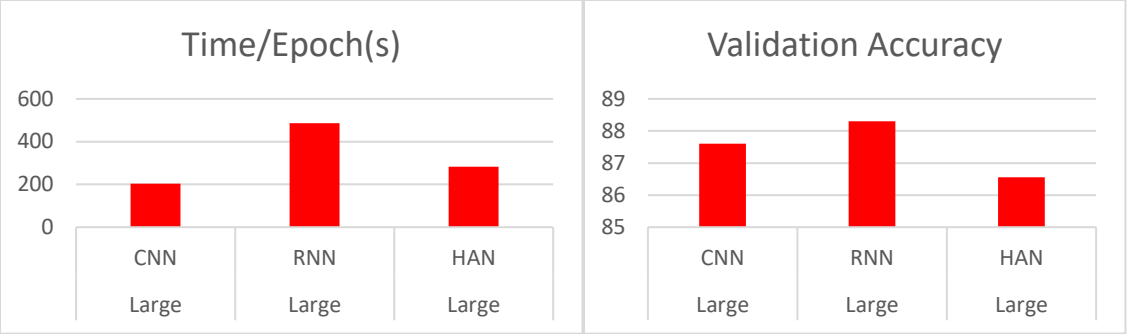
9. Time/Epochs Bar graph











10. Conclusion

The following observations are made based on the given dataset.

- Based on the above plots, RNN have achieved good validation accuracy with high consistency across all the datasets. HAN was able to achieve good accuracy on small dataset. Also, CNN has achieved high accuracy but that is not that consistent throughout all the datasets. In small size dataset, CNN shows only a maximum accuracy of 80%.
- CNN model has outperformed the other two models (RNN & HAN) in terms of training time, however RNN has consistently performed better than CNN and HNN.
- For different sized datasets, RNN seems to perform better in the given dataset, but in general, it depends on the dataset used and how we design our models.

11.References

- *https://www.nltk.org/api/nltk.tokenize.html*
- *https://keras.io/layers/writing-your-own-keras-layers/*
- *https://machinelearningmastery.com/develop-word-embedding-model-predicting-movie-review-sentiment/*
- *https://github.com/FlorisHoogenboom/keras-han-for-docla*
- *https://richliao.github.io/supervised/classification/2016/12/26/textclassifier-HATN/*
- *https://medium.com/jatana/report-on-text-classification-using-cnn-rnn-han-f0e887214d5f*