

Character Level Convolutional Networks for Text Classification

- Harish Umasankar
 - Ruhul Ameen
- Bodgam Rohan Reddy
 - Akash Vallamsetty

Goal

Use character convolution networks to classify text

Compare the accuracy of classification using the proposed model (using character level CNN) vs traditional method

Analysis on Impact of Various features on the Accuracy

Why do we need Character Level CNN

Traditional Method

- Quantizing by words to obtain features
- Issue: Number of words is very large

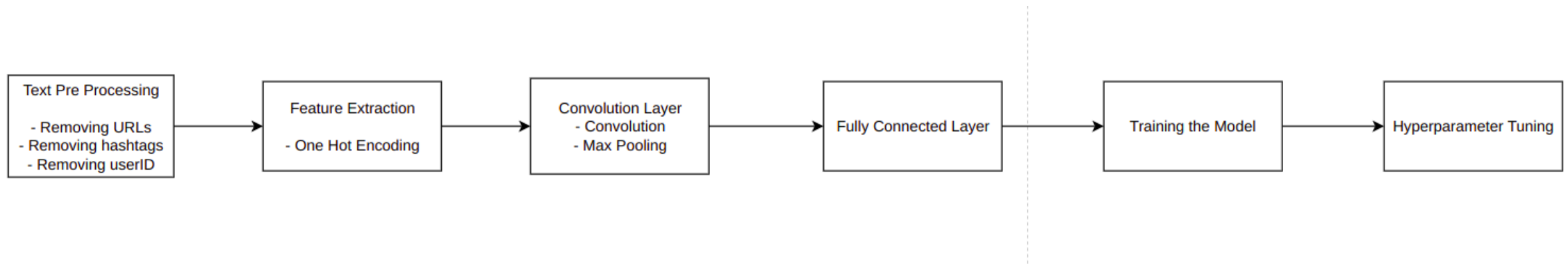
Proposed Method

- Total number of characters is limited
- Quantizing by character

Dataset: Tweets from Twitter

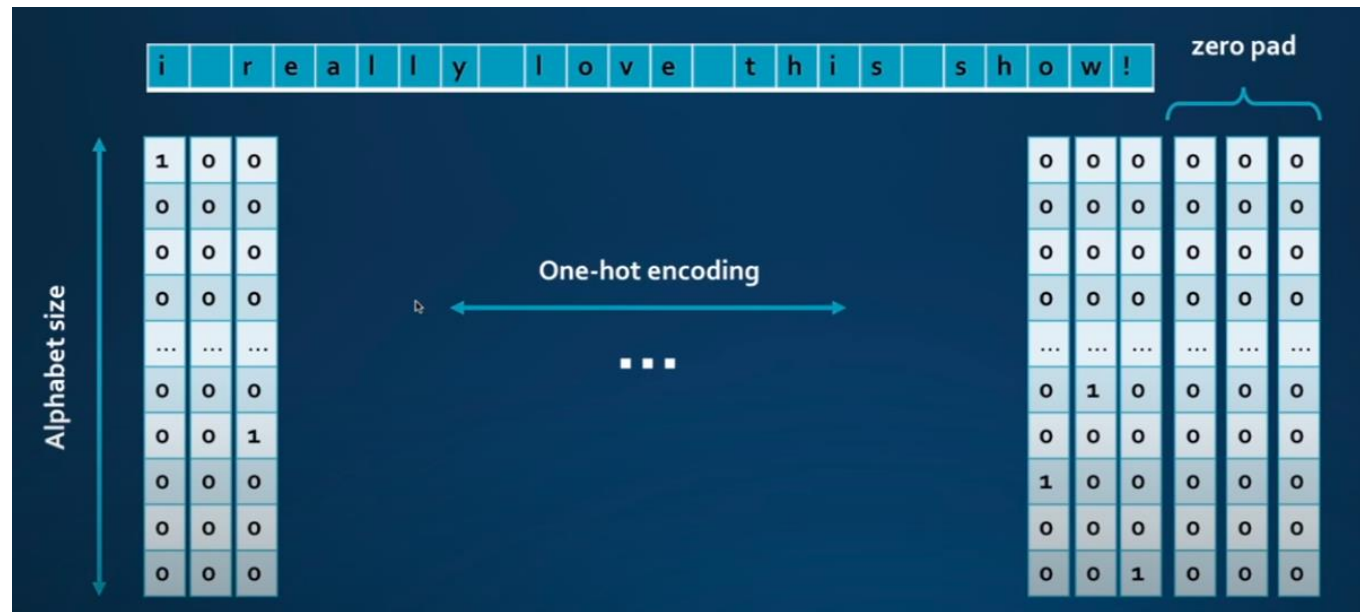
- Tweets are pulled from twitter. Number of tweets = 1.6 M
 - Tweets are tagged as positive and negative
 - Our goal use the proposed model in the paper to train the model to classify the tweets.
-
- Link to the [dataset](#)

Pipeline for Text Classification using Character Level CNN

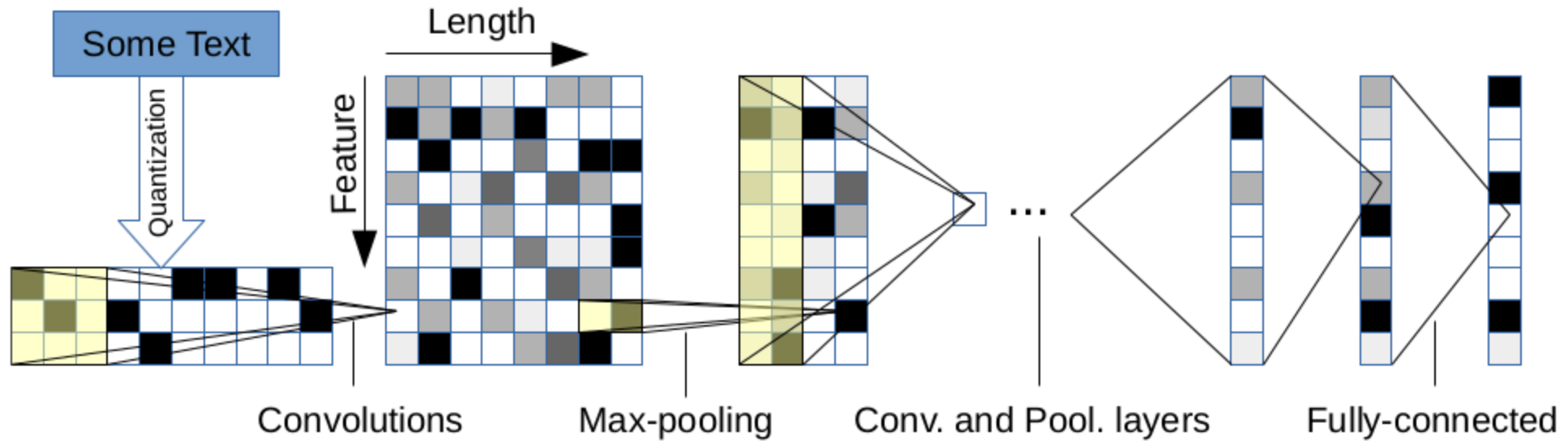


One Hot Encoding

```
abcdefghijklmnopqrstuvwxyz0123456789
- , ; . ! ? : ' ' ' / \ | _ @ # $ % ^ & * ~ ` + - = < > ( ) [ ] { }
```



Architecture



Character level CNN Logistics

- Quantize the string using one hot encoding
- Model consists of
 - 6 Convolution Networks
 - 3 Fully Connected Layers
- Convolutional Network
 - Strided Convolution

$$h(y) = \sum_{x=1}^k f(x) \cdot g(y \cdot d - x + c),$$

- Max-pooling function

$$h(y) = \max_{x=1}^k g(y \cdot d - x + c)$$

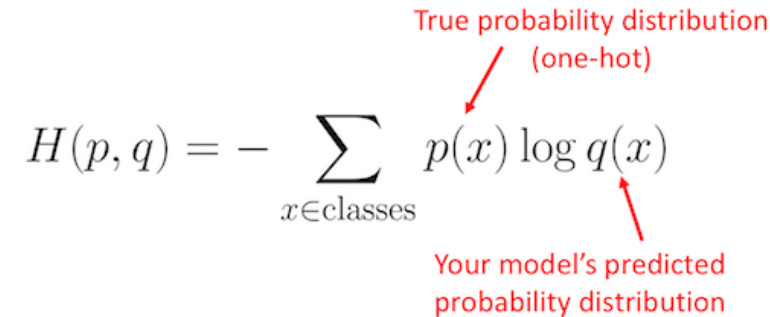
- Fully Connected Layer

Activation Function

- Used for classification based on the linear equation we will get after summing with weighted features
- Vanishing Gradient Problem in Sigmoid and Tan h Functions
- Why ReLu ?

Loss Function

- Used Sparse Categorical Cross Entropy Loss/Log Loss Function

$$H(p, q) = - \sum_{x \in \text{classes}} p(x) \log q(x)$$


True probability distribution
(one-hot)

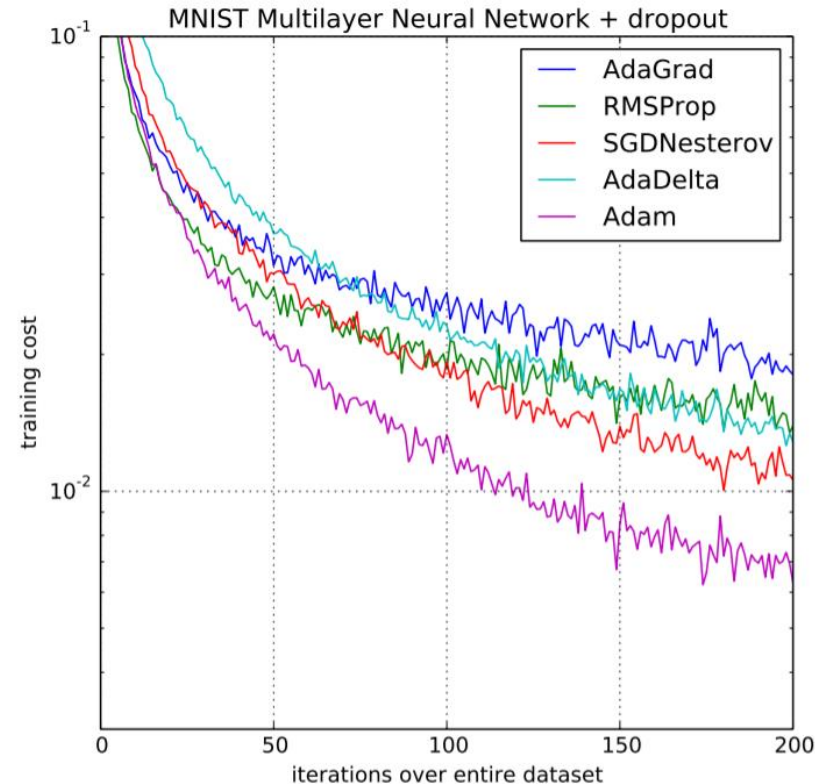
Your model's predicted
probability distribution

- Sparse Representation

In sparse categorical cross-entropy, truth labels are integer encoded, for example, [1], [2] and [3] for 3-class problem.

Optimizer

- Used Adam Optimizer
- Computes individual adaptive learning rates for different parameters



Result

- Dataset : 30 percent of whole split into 80-20 train-test data.
- Accuracy = 84%

Model	Precision	Recall	F1-measure	Accuracy
CNN(positive)	0.84	0.82	0.84	0.84
CNN(negative)	0.83	0.85	0.83	

Model 2: n Grams - TFIDF with Multinomial Logistic Regression

- Models constructed from selection of most frequent n-grams.
- N-gram : A sequence of n items from a given sample.

This is Big Data AI Book

Uni-Gram	This	Is	Big	Data	AI	Book
Bi-Gram	This is	Is Big	Big Data	Data AI	AI Book	
Tri-Gram	This is Big	Is Big Data	Big Data AI	Data AI Book		

- TFIDF is the method used to convert the n-gram words into feature vectors
- Multinomial Logistic Regression applied to the model to classify the test dataset and find accuracy.

Result

- Dataset : 30 percent of whole split into 80-20 train-test data.

Model	Precision	Recall	F1-measure	Accuracy
CNN (positive)	0.84	0.82	0.84	0.84
CNN (negative)	0.83	0.85	0.83	
N-gram TFIDF (positive)	0.76	0.80	0.78	0.78
N-gram TFIDF (negative)	0.79	0.75	0.77	

Model 3: Bag of Words TFIDF with Multinomial Logistic Regression

- Bag of words model collects/samples the most frequent words in the dataset.

the dog is on the table

0	0	1	1	0	1	1	1
are	cat	dog	is	now	on	table	the

- The frequencies of the words are converted into feature vectors using TFIDF.
- Multinomial Logistic Regression applied to the model to classify the test dataset and find accuracy.

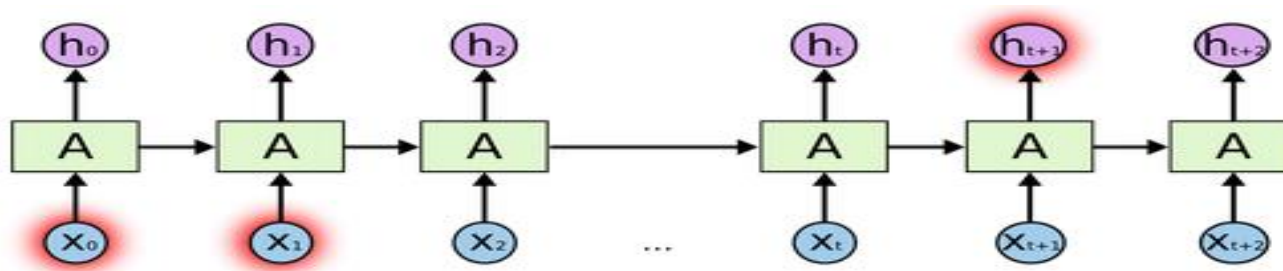
Result

- Dataset : 30 percent of whole split into 80-20 train-test data.

Model	Precision	Recall	F1-measure	Accuracy
CNN (positive)	0.84	0.82	0.84	0.84
CNN (negative)	0.83	0.85	0.83	
N-gram TFIDF (positive)	0.76	0.80	0.78	0.78
N-gram TFIDF (negative)	0.79	0.75	0.77	
Bag of words TFIDF (positive)	0.76	0.79	0.77	0.77
Bag of words TFIDF (negative)	0.78	0.75	0.76	

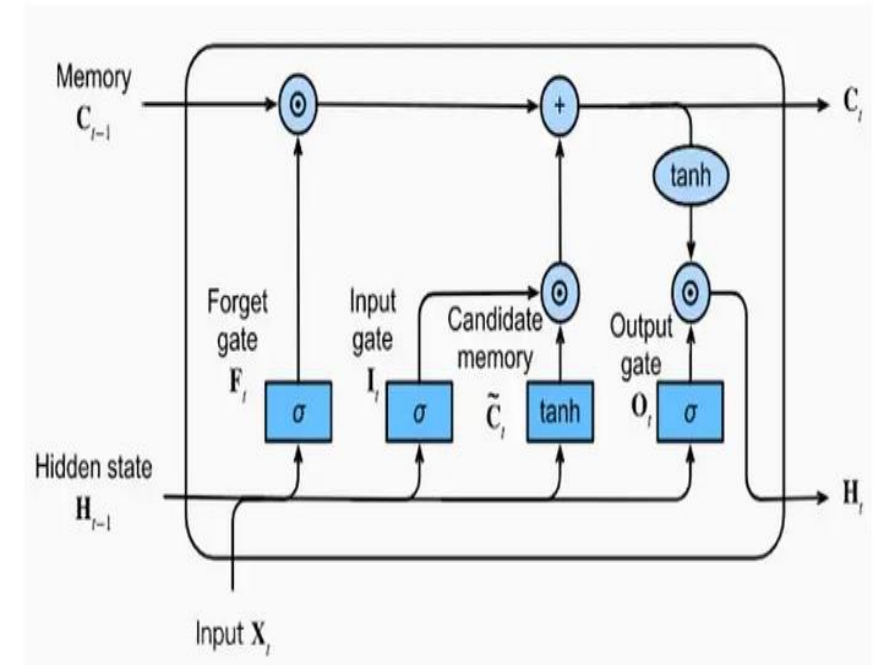
Model 4: LSTM

- LSTM is a special kind of Recurrent Neural Network(RNN)
- Normal RNN is not able to learn to connect information from a larger gap which is relevant
 - Eg. I grew up in France... So I speak fluent French . In this predicting French largely depends on the word 'France' which the RNN might not pick up.



Model 4:LSTM

- LSTM consists of two separate streams of memory – short term, long term
- It consists of three gates – Forget gate (decides how much long term memory to retain, input gate (updates long term memory), output gate (gives the output which is also next short term memory)
- The model is developed by introducing a LSTM layer in the deep neural network structure which gives the output as mean of all the LSTM cells present. The means are used as feature vectors and a classifier is developed using multinomial logistic regression.

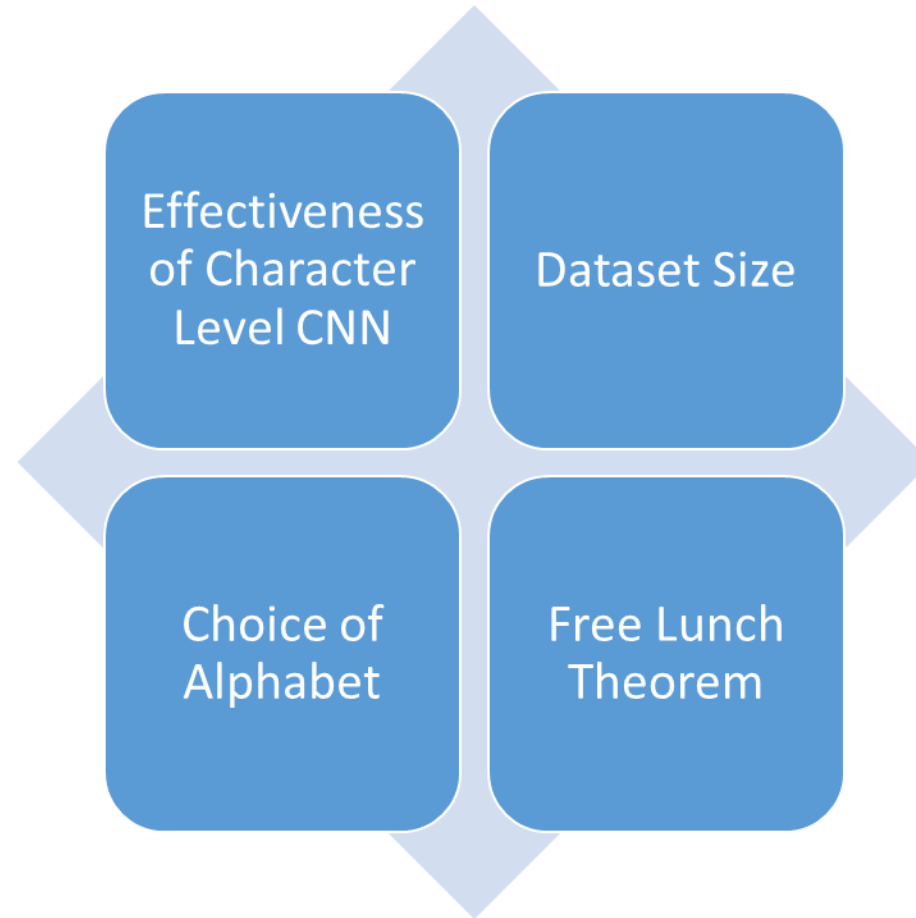


Result

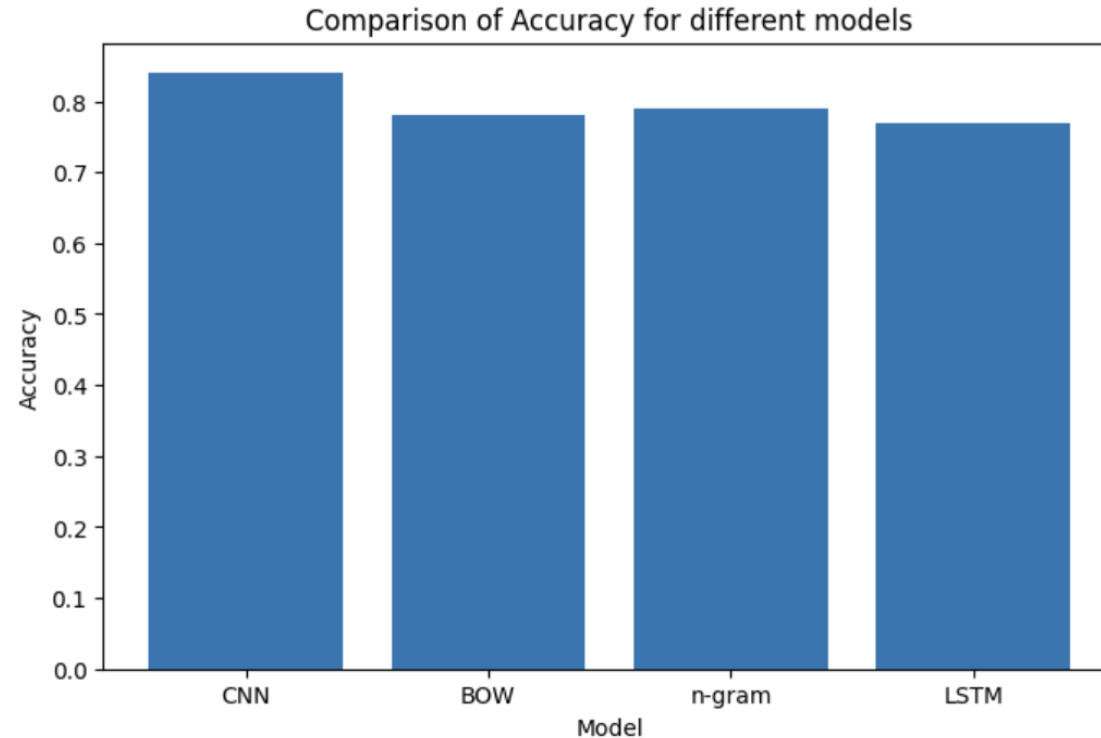
- Dataset : 30 percent of whole split into 80-20 train-test data.

Model	Precision	Recall	F1-measure	Accuracy
CNN (positive)	0.85	0.81	0.83	0.84
CNN (negative)	0.82	0.86	0.84	
N-gram TFIDF (positive)	0.78	0.81	0.79	0.79
N-gram TFIDF (negative)	0.80	0.77	0.78	
Bag of words TFIDF (positive)	0.77	0.80	0.78	0.78
Bag of words TFIDF (negative)	0.79	0.75	0.77	
LSTM (positive)	0.75	0.79	0.77	0.77
LSTM (negative)	0.78	0.74	0.76	

Analysis

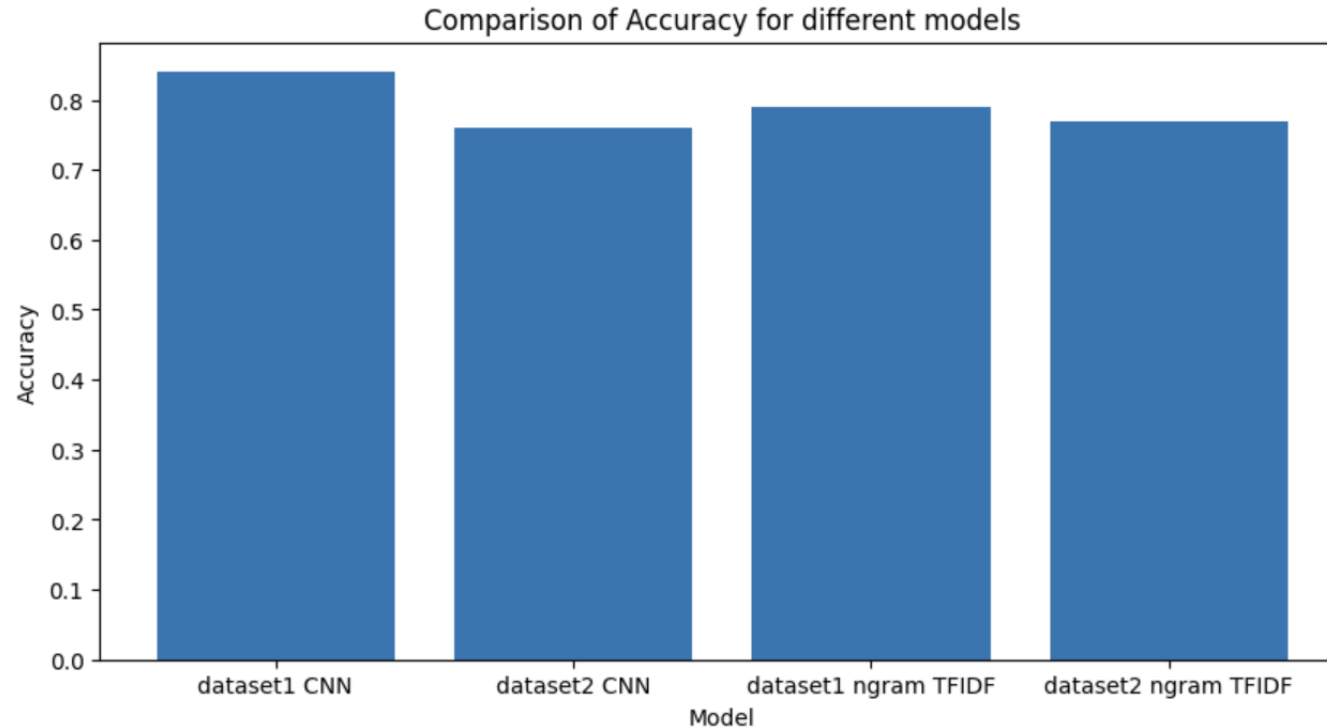


Effectiveness of Character Level CNN



- Perform text classification without need for words
- Text classification has more accuracy than other models

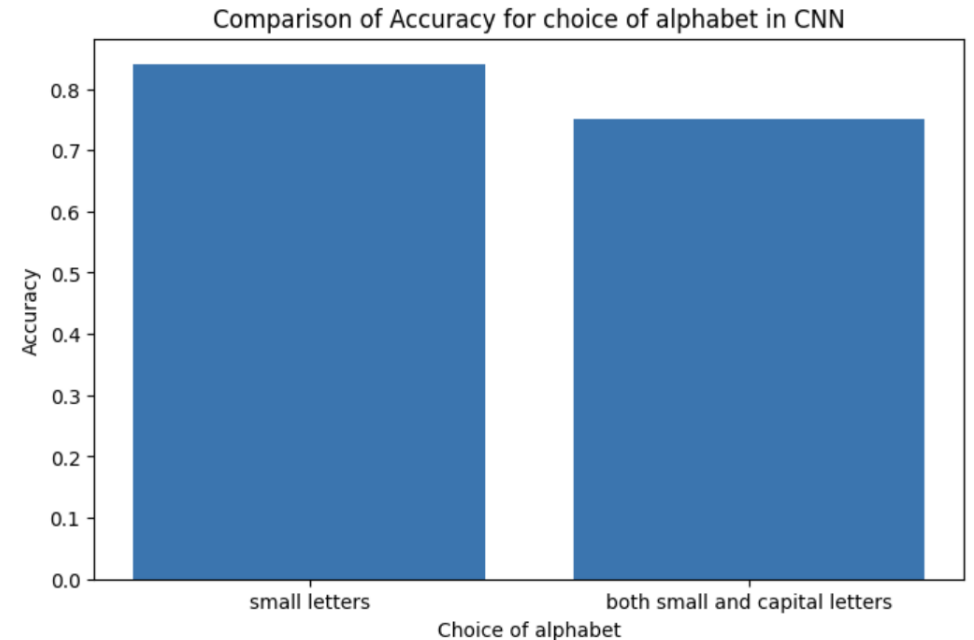
Dataset Size



- Convolution Networks work better on larger data sets
- Traditional Methods work better on smaller data sets

Choice of Alphabet

- Only using small letters, accuracy = 0.84
- Using both capital and small letters, accuracy = 0.75



- Making distinction between lower case and upper-case alphabets does not improve accuracy of model

Free Lunch Theorem

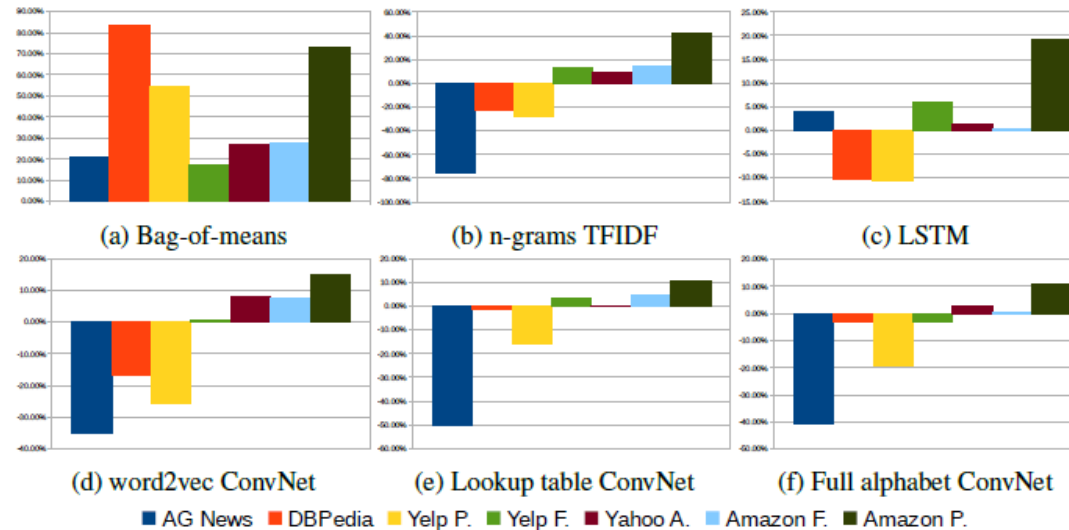


Figure 3: Relative errors with comparison models

- These experiments on Character Level CNN and Traditional Methods suggest that there no one single model that will work for all data sets

THANK YOU