Text Mining Project

Kangbo Chen(kanch152)

2022/1/12

Abstract

This report investigates whether state-of-art Convolutional Neural Networks(CNN) and Recurrent Convolutional Neural Networks(RCNN) outperform Machine Learning algorithms. The coverd Machine Learning algorithms include Naive Bayes, Logistic Regression, SVMs and Random Forest. Meanwhile, several vectorizars, such as CountVectorizer, tf-idf vectorizer, Word2Vec, GloVe,have been applied on different models, and different length of text data is also fitted to all models in order to get comprehensive comparison between models.(results and conclusion)

1 Introduction

Natural Language Processing domain has flourished for many years and a large number of papers have been published to develop new methodology solving NLP problems, such as text classification, searching, machine translation, part of speech tagging, etc. There are also various data sources, including social media, emails, user reviews, ticket website, and so on. Information carried by text data could be extremely rich, therefore, extracting insight from it has become the core issue.

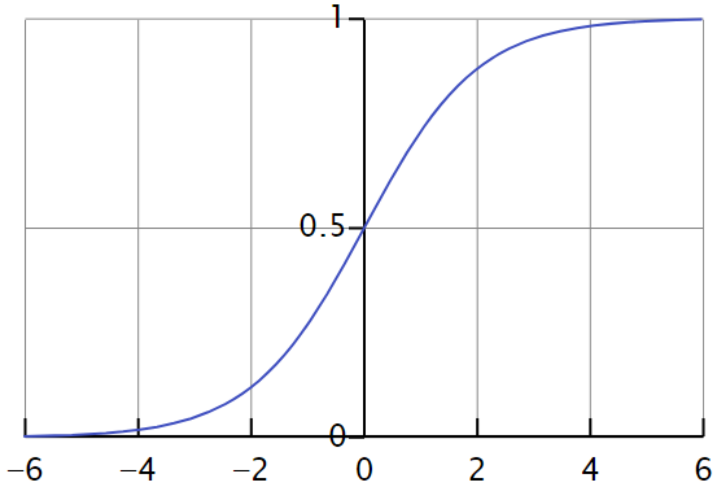
Due to the prosperity of artificial intelligence field and remarkable improvement of computational power, machine learning-based and deep learning-base algorithms have played an important role in NLP task, and text classification, as a classical and typical sub-field of NLP, where both types of algorithm have achieved a great success, attracts attention from many researchers around the world. For example, many machine learning algorithms, like naive bayes, logistic regression, decision tree, bagging and boosting algorithms, have been effectively applied on text classification first after linguistic approaches. Thanks to increasing computation power, deep learning-based model, inlcuding DNN, RNN, LSTM, CNN, ResNet, also start rising. Both method have pros and cons. Results form machine learning-based methods are more explainable, for instance, decision tree can tell which features are more important in classification, while deep learning model perform like a black box. In comparison, deep learning has a stronger potential to achieve better performance when size of dataset become huge, although it rely much more on hardware and is more time-comsuming.

Thus, this paper aims to compare performance of a series of Machine Learning algorithms with CNN(Convolutional Neural Networks) and RCNN(Recurrent Convolutional Neural Networks), two types of Deep Learning model, on text classification task. Various vectorizars, like CountVectorizer, tf-idf vectorizer, Word2Vec, GloVe, will be applied to evaluate the model as well.

2 Theory

2.1 Logistic Regression

Logistic Regression is named as regression, but is a practical discriminate model widely applied in classification problem, especially for binary classification. For example, the probability of true/false, positive/negative, win/lose labels is usually estimated using logistic regression.



The model of logistic regression is defined by following equation:

$$P(Y=1|X=x)=\frac{e^{x'\beta}}{1+e^{x'\beta}}$$

Where β is parameter vector for features and x is feature vector.

It can also be extended to multiclassification through softmax function replacing sigmoid function. The multiple version can be defined as following equation:

$$P(Y=j|X=x)=\frac{e^{x'\beta_j}}{\sum_{k=1}^{K}e^{x'\beta_k}}$$

Where β_k is the parameter vector for kth category.

2.2 SVMs

SVMs, stands for support vector machines, has become a standard state-of-art tool for pattern recognition in a very short period of time because of its outstanding performance. The idea for SVMs is to find a hyperplane maximizing upper boundary for different classes. For those dataset which cannot be seperated with a linear hyperplane, kernel method can be an alternative option to map the data into a higher dimensional space and do linear separation.

The goal of hard margin linear SVMs can be described as following equation:

$$\hat{w},\hat{b}=\underset{w,b}{argmax}\frac{1}{||w||},s.t.y_i(w'x_i+b)>1,i=1,2,...,m$$

where w represents parameter for hyperplane and b is bias

For nonlinear hyperplane, kernel function should be introduced and the object function can be described as following:

$$\hat{\lambda}=\underset{\lambda}{min}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda^{(i)}\lambda^{(j)}y^{(i)}y^{(j)}\kappa(x^{(i)},x^{(j)})-\sum_{i=1}^{N}\lambda^{(i)},s.t.\lambda^{(i)}>0,\sum_{i=1}^{N}\lambda^{(i)}y^{(i)}=0,i=1,2,...,N$$

where $\kappa(.)$ is kernel function

2.3 Multinomial Naive Bayes

There are six types of naive bayes model available in sklearn API, in this paper we use multinomial naive bayes specifically. Assume there ia text dataset $D = \{d_1, d_2, ..., d_n\}$, where $d_i(1 < i < n)$ refers to ith document and n refer the number of document. Let one feature set of D is $X = \{x_1, x_2, ..., x_m\}$ where x_i refers to a word in any document or other features ,like N-gram.

Given Bayes Theory:

$$P(Class|X) = \frac{P(X|Class)P(Class)}{P(X)}$$

What we should calculate are $P(X|Class)$,$P(Class)$,$P(X)$.

For $P(Class)$,

$$P(Class) = \frac{1}{||k||}$$

where k refers to the frequency of $class\,k$

For $P(X)$, it is a constant for each document and will not affect posterior distribution, therefore, here it can be ignored.

For $P(X|Class)$,

$$P(x_1, x_2, ..., x_m|Class\,i) = \prod_{j=1}^{m} P(x_j|c_i)$$

We use the assumption for naive bayes here that each feature is independent with each other.

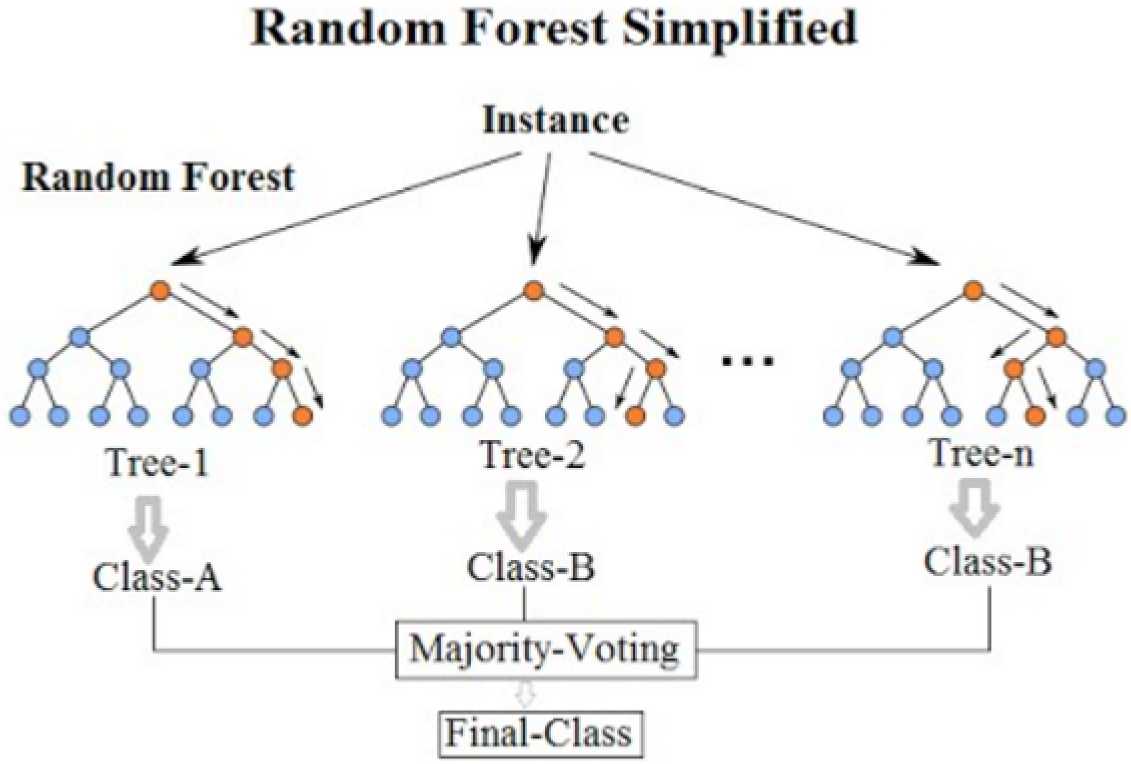
Putting the equations above together, we can get the simplified final equation for posterior probability.

$$P(Class|X) \propto P(c_i) \prod_{j=1}^{m} P(x_j|c_i)$$

Usually, logarithm is taken here to prevent computational collapse, and the chosen class is the one maximizing this probability.

2.4 Random Forest

Random Forest is a kind of ensemble method which combine the predictions of several basic estimators to improve robustness over single estimator.[sklearn]. The basic estimator for Random Forest refers to Decision Tree, a non-parametric supervised learning method. For each tree, a new set of samples drawn with replacement from training set is fitted, and all estimators are combined using average probability in sklearn API.



A best split node is found from either all input features or a random subset of presetting size has been defined by a parameter in sklearn API as fitting each tree.

2.5 CNN

Convolutional Neural Networks, known as CNN, has achieved remarkable results in the field of computer vision, such as image classification, object detection,etc, due to its strong ability to extract features. One the other hand, an increasing number of work making use of CNN for NLP has been involved and shown to be effective in recent years.

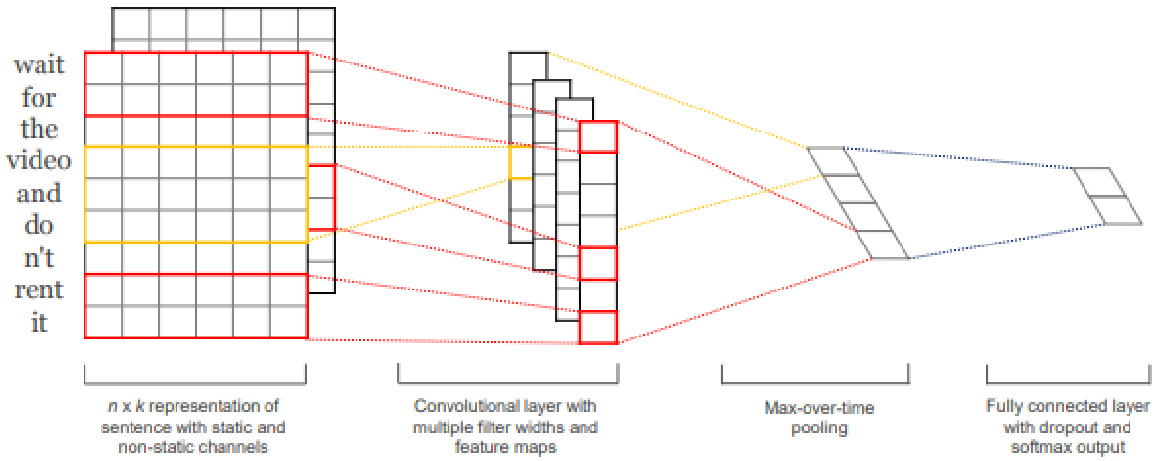
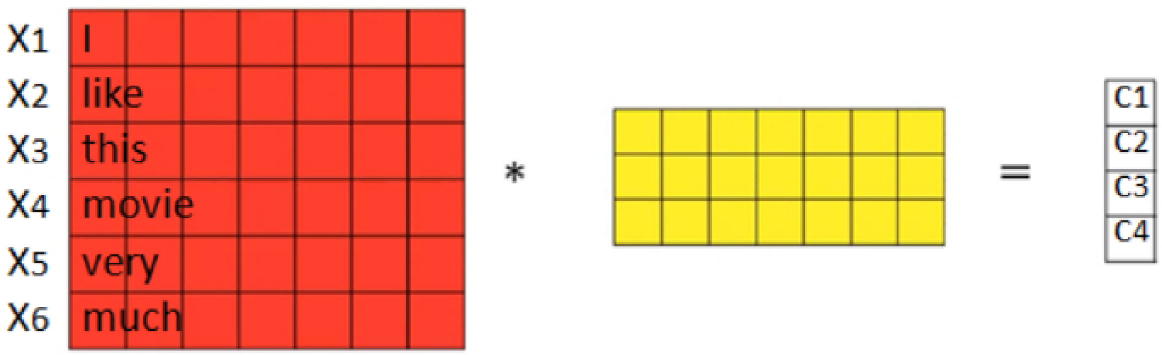

wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 1: Model architecture with two channels for an example sentence.

The model structure has been shown in Figure 1. It consists of several type of layers including input layer, convolutional layer, max-pooling layer and fully connected layer. The input layer requires the text data representation to be a fixed size matrix whose row dimension refers to the length of text and column is k-dimensional word vector for ith word in the text. The convolutional operation here is similar to that of image processing, but one-dimensional filter with certain window size is applied to generate new features. For instance, assuming $x_{i:i+h}$ represents the words within a h word window, a new feature c_i produced from this window can be expressed by Equation and Figure 2

$$c_i = f(W * x_{i:i+h-1} + b)$$



where W refers to a filter with size $h * k$, b is a bias term, and f is a non-linear activation function.

The window slides from top to the bottom and applies the operation on word vectors in the each window $\{x_{1:h}, x_{2:h+1}, ..., x_{n-h+1:n}\}$ to generate new feature

$$\boldsymbol{c} = (c_1, c_2, ..., c_{n-h+1})$$

The next step goes to max-pooling operation, which take the maximum value from output of each convolutional filter as selected feature,denoted as $\hat{c} = max(\boldsymbol{c})$, before enter the Dense layer to be finally classified.

Finally, a one hot output will be output after combined selected features being feed into fully connected neural networks, which can be described as following:

$$y = \sigma(H_n * \sigma(...(\sigma(H_1 * \hat{\boldsymbol{c}}) + b_1))) + b_n$$

where H_n is the weights to nth layer, b_n is bias,$\sigma(.)$ is activation function, y is one hot spot output,and $\hat{\boldsymbol{c}} = (\hat{c}_1, \hat{c}_2, ..., \hat{c}_j), j = the\, numbe\, of\, selected\, features$ coming from concatenated features after max-pooling.

2.6 RCNN

Compared with structure of CNN, we refer to recurrent structure, combining a word with its left and right context, which means we consider nearby words in the expression of each current word, not only see it as an independent sample. This structure helps us to capture a better meaning of current word. Let $c_l(w_i)$ be the left context of word w_i and $c_r(w_i)$ be the right context of word w_i and $e(w_i)$ be the word embedding of word w_i, then the left-side and right-side context can be expressed using following equations:

$$c_l(w_i) = f(W^{(l)} c_l(w_{i-1}) + W^{(sl)} e(w_{i-1}))$$

$$c_r(w_i) = f(W^{(r)} c_r(w_{i+1}) + W^{(sr)} e(w_{i+1}))$$

where $W^{(l)}$ is the matrix transforming the hidden context to the next hidden context and the matrix $W^{(sl)}$ represent the parameters integrating the semantic of current word with the next word's left context. f is a non-linear function. $W^{(r)}$ and $W^{(sr)}$ have the similar function with corresponding left-side matrix.

Then, let the representation of x_i being defined as

$$x_i = [c_l(w_i); e(w_i); c_r(w_i)]$$

which is the concatenation of left-side context,word embedding and right-side context vector.

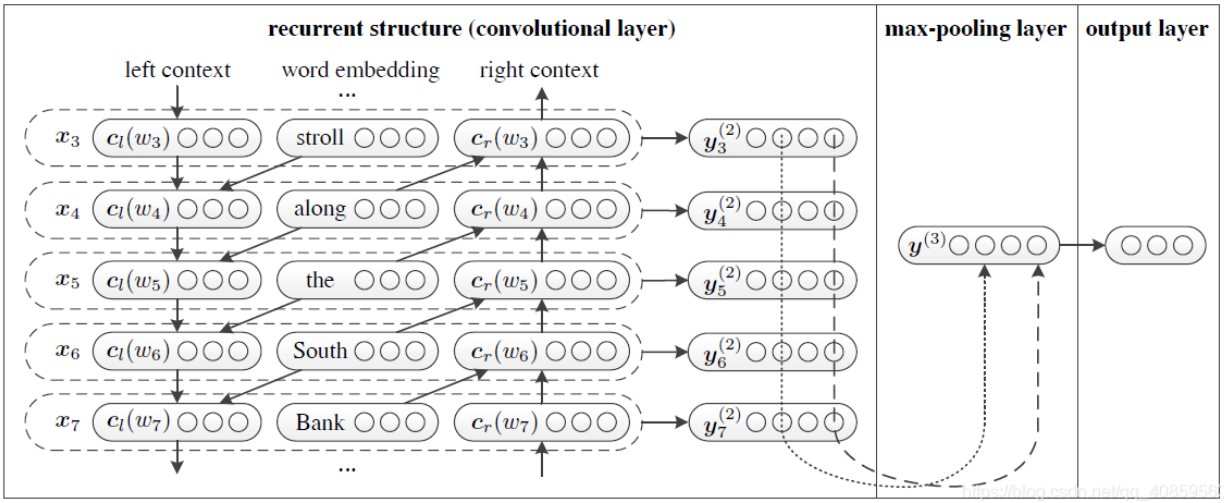
The next step in the structure figure is doing non-linear transformation using $tanh$ function on vector x_i, before feeding the output $y_i^{(2)}$ to max-pooling layer where the determination of useful factor is implemented.

$$y_i^{(2)} = tanh(W^2 x_i + b^{(2)})$$

After passing $y_i^{(2)}$ into max-pooling layer, we can get $y^{(3)}$

$$y^{(3)} = \max_{i=1}^{n} y_i^{(2)}$$

The last stage of our model is one or two dense layer which help us to do the final classification from selected features.



2.7 CountVectorizer and TF-IDF vectorizer

TF-IDF proposed by Sparck Jones(1972, 2004) developing from the definition of IDF. The basic idea behind is that

3 Datasets

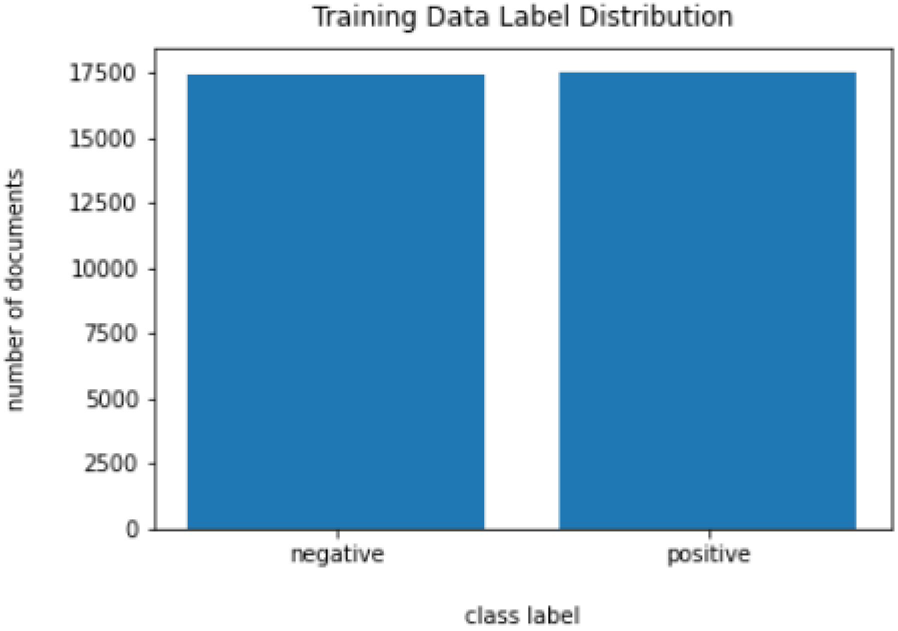
In order to determine the effectiveness of the models above, several datasets from kaggle have been used on which to compare the performance. These datasets are 'Amazon Reviews'(360k samples) held by J. McAuley and J. Leskovec. Hidden, 'IMDB Dataset of 50K Movie Reviews'(50k samples) held by Andrew Maas, 'Trip Advisor Hotel Reviews'(20k samples) held by Alam, M. H., Ryu, W.-J., Lee, S., 2016.

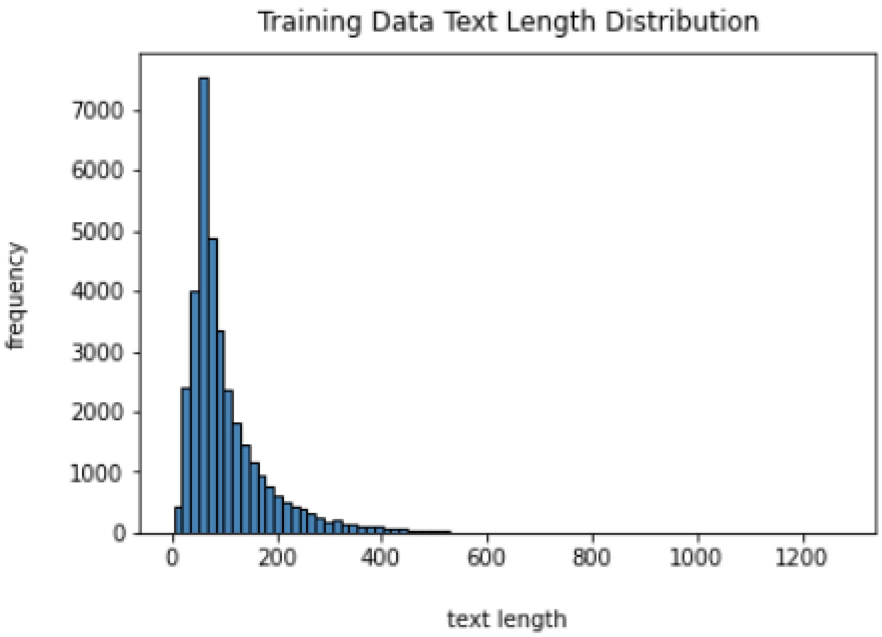
Amazon Reviews: The Original Amazon reviews dataset consists of reviews from amazon, including ~35 million reviews up to March 2013. Reviews include product and user information, ratings, and a plaintext review. This polarity sub-dataset is constructed by taking review score 1 and 2 as negative, and 4 and 5 as positive. Samples of score 3 is ignored. In the dataset, class 1 is the negative and class 2 is the positive. Each class has 1,800,000 training samples and 200,000 testing samples.

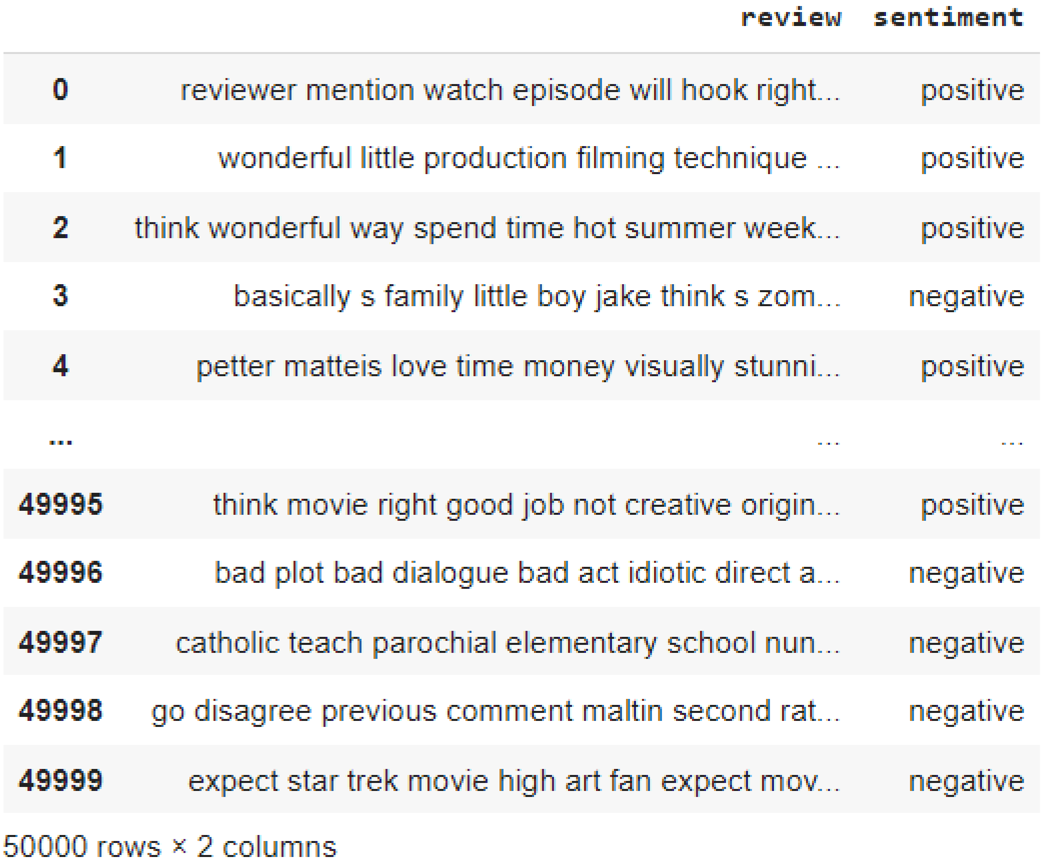
IMDB Dataset of 50K Movie Reviews: IMDB dataset having 50K movie reviews for natural language processing or Text analytics.

Trip Advisor Hotel Reviews: This dataset consisting of 20K reviews from Tripadvisor including unbalanced rates from 1 star to 5 stars.

Figure 4 illustrates the distribution of text length, and we can easily find the majority of comments stay less than 200 words, which can be helpful in choosing maximum sequence length for CNN.







	review	sentiment
0	reviewer mention watch episode will hook right...	positive
1	wonderful little production filming technique ...	positive
2	think wonderful way spend time hot summer week...	positive
3	basically s family little boy jake think s zom...	negative
4	petter matteis love time money visually stunni...	positive
...	...	...
49995	think movie right good job not creative origin...	positive
49996	bad plot bad dialogue bad act idiotic direct a...	negative
49997	catholic teach parochial elementary school nun...	negative
49998	go disagree previous comment maltin second rat...	negative
49999	expect star trek movie high art fan expect mov...	negative

50000 rows × 2 columns

4 Method

In this section, modification of each dataset is introduced and model parameters are also shown in more detail.

Since the 'Amazon Review' is quite large, there are several version of subset(0.2%, 1%, 10%, 100%) to study what the effect of size of data is. Besides, the heading column is also used to fit the models to compare performance on one sentence text. 'Trip Advisor Hotel Reviews' and IMDB dataset(has been tokenized and stored in 'preprocess_IMDB.pkl') are used to compare the influence of balanced and unbalanced data.

For Machine Learning model, there are many methods to do vectorization of original text, including word frequency, tf-idf, word2vector, pretrained vector. Here we use tf-idf vectorization only in order to simplify work flow.

For CNN model, 4 parallel convolutional layers, 2 dense layer is applied, the maximum sequence length(*maxlen*) for training and validation dataset is set to 200 due to the distribution of text length.

Result

Name	Academy	score
Harry Potter	Gryffindor	90
Hermione Granger	Gryffindor	100
Draco Malfoy	Slytherin	90

Discussion

Conclusion

References