ORIGINAL RESEARCH PAPER
Volume-7 | Issue-9 | September-2018 | PRINT ISSN No 2277 - 8179
INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH

TWITTER SENTIMENT ANALYSIS USING CAPSULE NETS AND GRU

Engineering

Maisnam Niranjan Singh Research scholar, Dayanand Sagar Engineering(VTU)

Y.S Kumaraswamy* Prof., Dean R & D(Nagarjuna Engineering College) *Corresponding Author

ABSTRACT

Sentiment analysis is one of the most researched areas in computer science. Opinion forming has been part of human behavior from the beginning but to teach a machine to conduct sentiment analysis is a hard problem solving task. The reason why it is hard for machine to do sentiment analysis stems from the fact that language could be very ambiguous . Sentiment analysis is all about understanding the language and classifying the language into something negative ,positive or neutral or like and not like. In this paper we tackle the sentiment analysis problem taken data from social networking site Twitter. In this paper we use deep learning approach to teach the machine to do sentiment analysis. We choose Capsule Net along with Bidrectional GRU , a form of recurrent neural network to do the sentiment analysis and use the Glove word embedding vectors . A case study is presented to illustrate and in depth study of the problem and we propose certain optimization steps to reach a high accuracy for the problem to be solved. Capsule Net a differs from the traditional convolution neural network which takes into account spatial correlation between the components whereas Convolution neural network cannot take spatial correlation, thus even though it works well there is still some disadvantage using convolution neural network.

KEYWORDS

1. INTRODUCTION

The rise of social media sites like Twitter, Facebook, Instagram have indeed generated so many data with so many people expressing their opinions on various topics. Some opinions in recent past have generated quite a flurry and has led to many discussions on whether these social networking sites are a boon to the society or not. Some tweets in Twitter could form positive or negative opinions which could lead to high consequences and could influence a decision making system. The recent US elections had generated quite a handful of tweets from both sides of the camp. Tweets could be analyzed and can determine whether who is going to win the presidential election from doing sentiment analysis on the tweets. Also there are many tweets generated from all around the people and by studying the tweets opinion can be formed and give quite useful information regarding terrorist sentiments which could help the government immensely.

Movie reviews, restaurant reviews and other reviews could really help others by those who have really experienced the journey by providing useful information such as a movie with a five star rating would generally turn out to be good , restaurant reviews can also help by giving information such as which dish will be palatable and which are not liked by many people.

This paper, studies the performances of two different architecture of modelling sentiment classification one LSTM and the other a combination of Capsule Net ,GRU and word embedding . Our result shows that the later , the combination of Capsule Net , GRU and word embedding gives better accuracy rate than LSTM

We provide the architecture of both LSTM and combined architecture of Capsule Net, GRU and word embedding and we also provide you code snippets of both the model build phase and model training phase of both , but due to other constraints we do not provide the full source code rather we provide only a few code to get the idea.

To help predict the sentiment we choose twitter data and collect and clean up the data so that these sentences can be fed into the neural network layer. We used Scikit-learn, a Python machine learning framework along with Pandas, Numpy, Keras, Tensorflow and Glove pretranied word embeddings.

We then compare the performance of different model architecture using metrics as accuracy score and choose the models having high accuracies. This whole idea of this paper is to show the performances of single LSTM and combined Capsule Net, GRU and word embeddings. Another idea is to show that using pre trained word vectors boost the performance of the accuracy of the models.

2. Previous Work

Previous work on sentiment analysis have used different models to predict the outcome. The different algorithms used are classical machine learning models like Gradient boosting, Random Forest and Support Vector Machines which have shown to give good accuracies but in this paper we will be using deep learning to perform sentiment analysis.

3.Data Types

Twitter data is has it own challenges as Twitter restricts the text in tweets to be no longer than 280 characters. Within these short texts opinions can be expressed in many forms. Since twitter data could be different on various topics from politics to recipe making , it is quite a big challenge which data to choose for sentiment analysis. Here we choose twitter data for an airline company and passengers travelling in airline to voice there opinions on what they think of the airline company. The opinions voiced by these passengers could be collected and depending on the sentiments voiced by these passengers , airline officials could take immediate action to rectify the problems in case if the sentiments expressed by the passengers show negative sentiments or if the sentiments of the tweets are positive the airline company can take extra steps to increase efficiency and avoid dissatisfied passengers. On the other hand , passengers could form an opinion which airline to choose to travel based on the opinions voiced by other co passengers and have a safe and enjoyable journey.

The data contains 10 columns which is given in the below format. Sentiment could be either positive , negative or neutral.

ID	Tweet	Retweet	Likes	Reason	company	Create time	location	time zone	sentiment
234	@Airlines Nice service	24	4	Comfortable	Z airline	3/20/2015 10:35	Boston	Pacific Time	postive
234	@SuperAirlines Uncomfortable experience	34	65	Seat was not adjustable	super airline	3/21/2015 1:35	San Francisco	Pacific Time	negative
234	@VirginAirlines Need to sleep	1	0	It will take 2 hours	Z airline	3/11/2015 12:35	New Yort	Pacific Time	neutral

FIGURE :1

4.Feature Engineering

As part of feature engineer we need to transform the text into word embedding vectors and so we use Glove which is publicly available from Stanford site . The text in the tweet is converted to vectors and retweet, likes,time zone , date, reason, location are all taken into account and feature engineered to convert it to numbers and the date field is converted into day, month, year and time.

Once all the field are converted , standard scaling is performed so all the data is on the same scale we feed these processed data into the different layers of neural network.

5. LSTM implementation .

```
import numpy as np
import pandas as pd
import re

from sklearn.feature_extraction.text import CountVectorizer
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences

def loadAndclean():
    data = pd.read_csv('train.csv')
    data['text'] = data['text'].apply(lambda x: str(x).lower())
    data['text'] = data['text'].apply((lambda x: re.sub('[^a-zA-z0-
9\s]','',str(x))))
    return data

max_words = 500
data = loadAndclean ()
tok = Tokenizer(nb_words=max_words, split=' ')
tok.fit_on_texts(data['text'].values)
X = tok.texts_to_sequences(data['text'].values)
X = pad_sequences(X)
```

```
from keras.models import Sequential
from keras.layers import Dense, Embedding, LSTM

embed_dim = 32
lstm_out = 10
def buildModel():
    model = Sequential()

model.add(Embedding(max_words,embed_dim,input_length=X.sha
pe[1]))
    model.add(LSTM(lstm_out))
    model.add(Dense(3, activation='softmax'))
    model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
    return model
```

```
model = buildModel()
model.fit(X_train,Y_train, nb_epoch=4, batch_size=1, verbose=1)

Epoch 1/4
2237/2237 [==============================] - 432s
193ms/step - loss: 0.4445 - acc: 0.7976
Epoch 2/4
2237/2237 [==============================] - 416s
186ms/step - loss: 0.3713 - acc: 0.8285
Epoch 3/4
2237/2237 [==============================] - 398s
178ms/step - loss: 0.3290 - acc: 0.8520
Epoch 4/4
2237/2237 [==============================] - 438s
196ms/step - loss: 0.2849 - acc: 0.8806
```

Figure 2

6. Capsule Net

Once the data processing step is over , the data is processed further by cleaning and special characters are removed from the text and then multiple models were built on the data sets using various parameter settings. Capsule Net architecture which contains capsules and dynamic routing was recently published by Geoffrey Hinton , one of the pioneers of deep learning has a totally different take from the conventional Convolutional Neural Network which is used as one of the most successful Neural network for image classification and video classification. CNN even though are very successful and already have a large number of publicized papers and implementation from one serious drawback. CNNS do not take into account the spatial relationship between the various components. The better we the neural network understand the relationship between the different components the better understanding between the relationships where as CNN complete ignore this step.

7. GRU

GRU, is a type of recurrent neural network , which is used in sequence modeling problems and are a perfect fit for natural language processing task. It is a simpler version of LSTM and solves the vanishing gradient problem. Neural networks learn by using Gradient descent and in order for the neural network to learn during the phase of back propagation , the gradients are calculated and GRU as in LSTM tries to remember the state of the previous step by keep the gradient to not fall to zero as this would mean a partial derivation of a zero would always be zero and hence the neural network would refuse to learn anything.

8. Architecture

In order to perform the sentiment analysis we propose an architecture of Capsule Nets combined with Bidirectional GRU to obtain the best results. The deep learning framework we use is Keras with Tensorflow as the Backend and the Glove pre trained word embedding vectors . These three combined together processes the data to predict the sentiment .

The data set which contains 11 columns are featured engineered differently. The data field is converted to day, month, year, time, the other text are converted to word embedding vector using Glove. After the conversion is done the data is fed into the Capsule Net and then the capsules processes the data and then is fed further to the Bidirectional GRU layer which then runs for a few epochs and then the sentiment is calculated into either 1 being position , 0 being neutral and 1 being negative.

Step 1	Step 2	Step 3	Step 4	Step 5
Feature Engg	Embedding	GRU	Capsule	Prediction

FIGURE 3

9. Word Embedding matrix

We will be showing only the important part of the code , the tokernizer, the model architecture and the training phase The tokenizer code in python is given below

```
word_index = tokenizer.word_index
#prepare embedding matrix
num_words = min(max_features, len(word_index) + 1)
embedding_matrix = np.zeros((num_words, embed_size))
for word, i in word_index.items():

    if i >= max_features:
    continue

    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
    # words not found in embedding index will be all-zeros.
    embedding_matrix[i] = embedding_vector
```

FIGURE 4

Below is the python function which builds the model, first the raw text data is converted to embedding matrix using the Glove pre trained vectors and as we are using Keras the input is fed into the Embedding Layer of Keras which then is passed to Bi-directional GRU layer which is later on fed into the Capsule Net class.

```
def get_model():
    inp = Input(shape=(maxlen,))
    x=Embedding(max_features, embed_size,
weights=[embedding_matrix],trainable=False)(inp)
    #embed_layer = Embedding(max_features,
                    #embed_size,
 #input_length=maxlen,
                    #weights=[embedding_matrix],
                    #trainable=False)(input1)
    x = SpatialDropout1D(rate_drop_dense)(x)

    x = Bidirectional(
      GRU(gru_len, activation='relu', dropout=dropout_p,
recurrent_dropout=dropout_p, return_sequences=True))(x)
```

```
x = Capsule(num_capsule=Num_capsule,
dim_capsule=Dim_capsule, routings=Routings,
                    share_weights=True)(x)
    # output_capsule = Lambda(lambda x:
K.sqrt(K.sum(K.square(x), 2)))(capsule)
    x = Flatten()(x)
    x = Dropout(dropout_p)(x)

    output = Dense(3, activation='sigmoid')(x)
    model = Model(inputs=inp, outputs=output)
    model.compile(
        loss='binary_crossentropy',
        optimizer='adam',
        metrics=['accuracy'])
    #model.summary()
    return model

model = get_model()

batch_size = 32
epochs = 5
X_tra, X_val, y_tra, y_val = train_test_split(x_train, Y,
train_size=0.85, random_state=2018)
RocAuc = RocAucEvaluation(validation_data=(X_val, y_val),
interval=1)
hist = model.fit(X_tra, y_tra, batch_size=batch_size, epochs=10,
validation_data=(X_val, y_val),
                callbacks=[RocAuc])
loss, acc = model.evaluate(Xtest, ytest, verbose=0)
print('Test Accuracy: %f' % (acc*100))
y_pred = model.predict(x_test, batch_size=32, verbose=1)
y_classes = y_pred.argmax(axis=-1)
```

FIGURE 5

10. Results and Discussion

In generally almost all the models have not given extraordinary accuracy results, but Random Forest Classifier was chosen as the best model after multiple tests. An AUC score of 0.73 given by Random Forest model.

hist = model.fit(X_tra, y_tra, batch_size=batch_size, epochs=10, validation_data=(X_val, y_val),

callbacks=[RocAuc])

```
Train on 2838 samples, validate on 501 samples

Epoch 1/10
2838/2838 [==============================] - 41s
14ms/step - loss: 0.0899 - acc: 0.9663 - val_loss: 0.1097 - val_acc:
0.9534

 ROC-AUC - epoch: 1 - score: 0.982051
Epoch 2/10

2838/2838 [==============================] - 98s
35ms/step - loss: 0.0836 - acc: 0.9676 - val_loss: 0.1117 - val_acc:
0.9554

 ROC-AUC - epoch: 2 - score: 0.982780

Epoch 3/10
2838/2838 [==============================] - 41s
14ms/step - loss: 0.0807 - acc: 0.9677 - val_loss: 0.1115 - val_acc:
0.9594

 ROC-AUC - epoch: 3 - score: 0.981609

Epoch 4/10
2838/2838 [==============================] - 84s
30ms/step - loss: 0.0815 - acc: 0.9697 - val_loss: 0.1123 - val_acc:
0.9601

 ROC-AUC - epoch: 4 - score: 0.980708

Epoch 5/10
2838/2838 [==============================] - 64s
23ms/step - loss: 0.0707 - acc: 0.9733 - val_loss: 0.1207 - val_acc:
0.9528
```

```
 ROC-AUC - epoch: 5 - score: 0.981838

Epoch 6/10
2838/2838 [==============================] - 72s
26ms/step - loss: 0.0715 - acc: 0.9744 - val_loss: 0.1303 - val_acc:
0.9508

 ROC-AUC - epoch: 6 - score: 0.980494
Epoch 7/10
2838/2838 [==============================] - 69s
24ms/step - loss: 0.0658 - acc: 0.9766 - val_loss: 0.1244 - val_acc:
0.9601

 ROC-AUC - epoch: 7 - score: 0.978591
Epoch 8/10
2838/2838 [==============================] - 73s
26ms/step - loss: 0.0636 - acc: 0.9764 - val_loss: 0.1229 - val_acc:
0.9574

 ROC-AUC - epoch: 8 - score: 0.979026
Epoch 9/10
2838/2838 [==============================] - 69s
24ms/step - loss: 0.0531 - acc: 0.9827 - val_loss: 0.1263 - val_acc:
0.9581

 ROC-AUC - epoch: 9 - score: 0.979734
Epoch 10/10
2838/2838 [==============================] - 70s
25ms/step - loss: 0.0505 - acc: 0.9827 - val_loss: 0.1342 - val_acc:
0.9554

 ROC-AUC - epoch: 10 - score: 0.980280
```

FIGURE 6

As we can see after running 10 epochs we achieve a score of 0.980280 on the training data and a score of 0.9554 on valiation.

11. Performance improvement using different architecture

For most of the sequence modelling recurrent neural networks has been used , but recurrent neural network can learn sequences upto a few sequences as the gradient becomes smaller and smaller, the network refuses to learn and the improvement stops. To alleviate these problems of vanishing gradient LSTM and GRU have been proposed for learning sequences which could remember long sequences than the recurrent neural network counter part.

To improve performances different deep learning architecture could be used, hyper paramenters could be selected and selected and tuned. Different optimizers could be used and the learning rate could be adjusted and different iterations could be experimented.

We believe that using different architecture, using dropouts to avoid overfitting, using Batch Normalization and tuning of hyper parameters could help boost the performance of sentiment analysis.

12. Conclusion and Future Work

This paper has shown that using Bi-directional GRU , Capsule Nets and Glove pre trained word outperforms LSTM without word embeddings .The accuracy score of LSTM without using pre trained word embedding is around 0.8806 and the combined architecture Bi-directional GRU , Capsule Nets and Glove pre trained word achieves a score: 0.980280 of for performs better in terms of accuracy. But there are other architecture which could have been used such LSTM, CNNS together

REFERENCES

[1] https://keras.io/
[2] https://keras.io/layers/recurrent/#gru
[3] https://keras.io/layers/recurrent/#lstm
[4] https://keras.io/layers/embeddings/
[5] https://nlp.stanford.edu/projects/glove/ 2012.
[6] https://www.tensorflow.org/guide/
[7] Kim, S.-M., and Hovy, E. 2004. Determining the sentiment of opinions. In Proceedings of Coling.
[8] Hatzivassiloglou, V., and McKeown, K. 1997. Predicting the semantic orientation of adjectives. In Proc. of ACL.
[9] Dave, K., Lawrence, S., & Pennock, D. M. (2003). Mining the peanut gallery: opinion extraction andsemantic classification of product reviews. In Proceedings of the 12th international WWW conference, May 20–24, 2003 (pp. 519–528). Budapest, Hungary.
[10] Gamon, M. (2004). Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In Proceedings of the 20th international conference on computational linguistics (COLING 2004), August 23 – 27, 2004 (pp. 841–847). Geneva, Switzerland.
[11] Dynamic Routing Between Capsules Sara Sabour, Nicholas Frosst, Geoffrey E. Hinton Published 2017 in NIPS

[12] Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling Junyoung Chung, Çaglar Gülçehre, +1 author Yoshua BengioPublished 2014 in ArXiv

[13] Gated Feedback Recurrent Neural Networks Junyoung Chung, Çaglar Gülçehre, +1 author Yoshua BengioPublished 2015 in ICML

[14] Hiroshi, K., Tetsuya, N., & Hideo, W. (2004). Deeper sentiment analysis using machine translation technology. In Proceedings of the 20th international conference on computational linguistics (COLING 2004),August 23 – 27, 2004 (pp. 494–500). Geneva, Switzerland.

[15] Tumasjan, A.; Sprenger, T. O.; Sandner, P.; and Welpe, I. 2010. Predicting elections with twitter: What 140 characters reveal about political sentiment. In Proceedings of ICWSM.

[16] Yu, H., and Hatzivassiloglou, V. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In Proc. Of EMNLP.

[17] Barbosa, L., and Feng, J. 2010. Robust sentiment detection on twitter from biased and noisy data. In Proc. of Coling.

[18] Jansen, B. J.; Zhang, M.; Sobel, K.; and Chowdury, A. 2009. Twitter power: Tweets as electronic word of mouth. Journal of the American Society for Information Science and Technology 60(11):2169–2188. [19] Pang, B., and Lee, L. 2008. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2):1–135.

[20] S. Batra and D. Rao, ”Entity Based Sentiment Analysis on Twitter”, Stanford University,2010

[21] I. Feinerer, Introduction to the tm Package Text Mining in R. 2015.

[22] K. Dave, S. Lawrence, and D. M. Pennock, “Mining the peanut gallery: Opinion extraction and semantic classification of product reviews,” in Proceedings of the 12th international conference on World Wide Web, 2003, pp. 519–528.

[23] B. Liu, “Handbook Chapter: Sentiment Analysis and Subjectivity. Handbook of Natural Language Processing,” Handbook of Natural Language Processing. Marcel Dekker, Inc. New York, NY, USA, 2009.

[24] R. Parikh and M. Movassate, “Sentiment Analysis of User-Generated Twitter Updates using Various Classification Techniques”, CS224N Final Report, 2009

[25] Pang, B. & Lee, L. (2004). A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of the 42nd annual meeting of the Association for Computational Linguistics (ACL), July 21–26 , 2004 (pp. 271–278). Barcelona, Spain.

[26] Ensemble Learning Better Predictions Through Diversity by Todd Holloway ETech 2008

[27] Witten, I. H. & Frank, E. (2005). Data mining: practical machine learning tools and techniques. Morgan Kaufmann, San Francisco, 2nd edition.