Iran University of Science and Technology
Deep Learning (Spring 2020)

Sentiment Analysis Using CNN-LSTM Based on Emoji-Sense

Maryam Sadat Eslami
Department of Computer Engineering
Iran University of Science
and Technology
maryam_eslami@comp.iust.ac.ir

Abstract

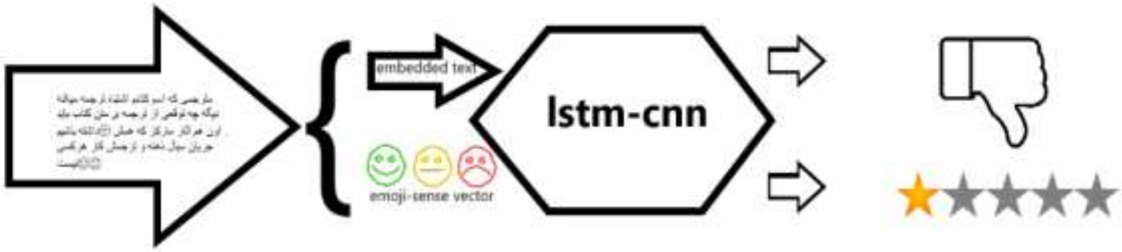
Semantic Analysis is a process of identifying whether the writer's attitude is positive or negative. By increasing user's interactions in social network, there is a great opportunity to gather data and do analysis for classification, prediction or knowledge extraction. In this classification NLP task, we propose a Bidirectional Long Short Term Memory(Bi-LSTM) and a Convolutional Neural Network(CNN) model with embedded text and emoji-sense vector as inputs to predict whether a comment's opinion is positive or negative and predict 5-star rating. Emojis helps better understanding of sense behind a piece of text. The proposed approach, combination of CNN and LSTM, is based on emoji-sense vectors.

Keywords: Long Short Term Memory, Semantic Classification, Social Network Analysis, Word Embedding, Emoji Sense, Natural Language Processing, Deep Learning

1 Introduction

Semantic analytics help us to monitor and analysis user's emotion and opinion. Analyzing user's comments on every production, improves digital marketing and helps to know customer's taste. As a result of limited length and lack of context in comments, analysis is challenging. Furthermore, Emoji play an important role to identify whether a comment's opinion is positive or not. Embedded text and emoji-vector are fed to output of CNN and LSTM parts and after some dense and average layers, model make prediction.

In this classification task, user's 5-star rating are labels. Input is user's comments and Two outputs of model are 5-star rating and being recommended or not (a binary classification).



model baseline

First step of data representation after preprocessing is text embedding. A mathematical representations of words as vectors in a way that similar words' vectors have less distance than unrelated or opposite words. Embedded vectors are input of the network. Given short comment's may have not enrich features. Assign an emoji-sense vector to each comment, which contain emoji sense in positive, neutral and negative mood, helps to identify sentence sense. Comments without emojis concatenate with zero vector.

As a result of having two outputs, two labels are prepared. 5-star rating and being recommended or not (a binary classification). For the first output onehot encoded vectors are represented and for second output, one or zero for rating more or less than 3 of 6.

As feedforward networks, CNNs can learn local features from sentences While RNNs memorize temporal dependencies in sequential data. LSTM an improved version of RNN, which with feedback connections deal with vanishing gradient problem, remembers long-term relations. In our task Bi-LSTM is a better choice because association of positive words or negative words is considered and ordering does not matter. Ensemble of CNN and Bi-LSTM helps to improve model accuracy. In this model 1-dimentional CNN and Bi-LSTM sub models are used.

2 Related work/Background

Sentiment analysis is a common task in NLP area. researchers have used different types of sentiment analysis techniques such as lexicon based and machine learning. Lexicon methods calculate orientation for a document from the semantic orientation of words or phrases in the document refers to a real number measure of the positive or negative sentiment expressed by a word or phrases [1] [2] [3] [4]. Machine learning methods use some ML approaches like SVM [5]. Also Learning Word Vectors for Sentiment Analysis and multi representing, supervised technique for sentiment similarity and unsupervised for semantic similarities [6].

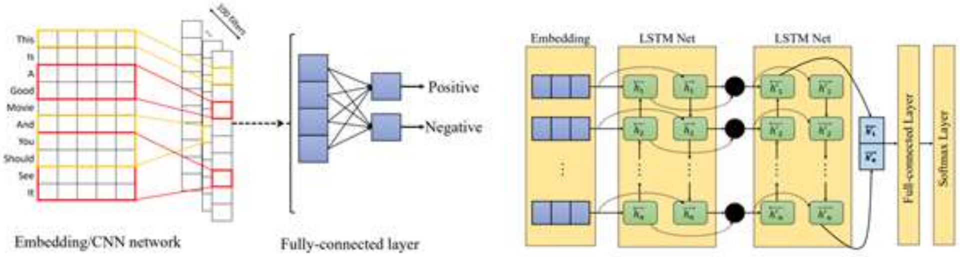
Effective representing text for deep neural network is a critical component. TFIDF, bag of words (is very high dimensional), pretrained word embedding like Word2Vec, etc. are some old techniques. SACPC [7] is a novel none deep approach for short text like comments which uses probabilistic linguistic terms(PLTSs) and support vector machines(SVM) for sense classification. State of the art of text processing is transformers. Using transformers for word encoding is another form of word representation which is then fed to Bi-LSTM with attention to determine the sentiment [8]. Bidirectional encoder representations from transformers (BERT), sentiment-specific word embedding models, cognition-based attention models, common sense knowledge, reinforcement learning, and generative adversarial networks are other approaches to solve the problem.

Emoji embedding is a novel approach to learn emoji embedding under positive and negative sentimental tweets individually, and then train a sentiment classifier by attending on these bi-sense emoji embedding with an attention-based long short-term memory network (LSTM) [9].

3 Proposed method

As a preprocessing step, emojis are removed from text and emoji-sense vectors are made as another single input. Text normalization is the process of transforming text into a single canonical form (for example corrects half spaces). Lemmatization, Stemming and removing stop words lead to root tokens of meaningful words for processing. Preprocessed data fed to embedding layer to be represented in vectors with constant length. Emoji-vectors are made from Emoji sentiment data[1]. The sentiment is computed from 70,000 tweets, labeled by 83 human annotators in 13 European languages. Two text vector and emoji-sense vector fed to model in different steps.

A stack of Bi-LSTM layers remembers association of words not focus on sequence order. Series of 1-dimentional CNNs lead to gather local features. Emoji-vectors concatenate with output of both CNN and LSTM segments. A fully-connected output layer that maps the CNN and LSTM layer outputs to a desired output size. A sigmoid activation layer represents being recommended or not, and the softmax layer predicts 5-star rating. Both final outputs are average of CNN and LSTM outputs. Dropout layers lead to less overfitting and more generalization.



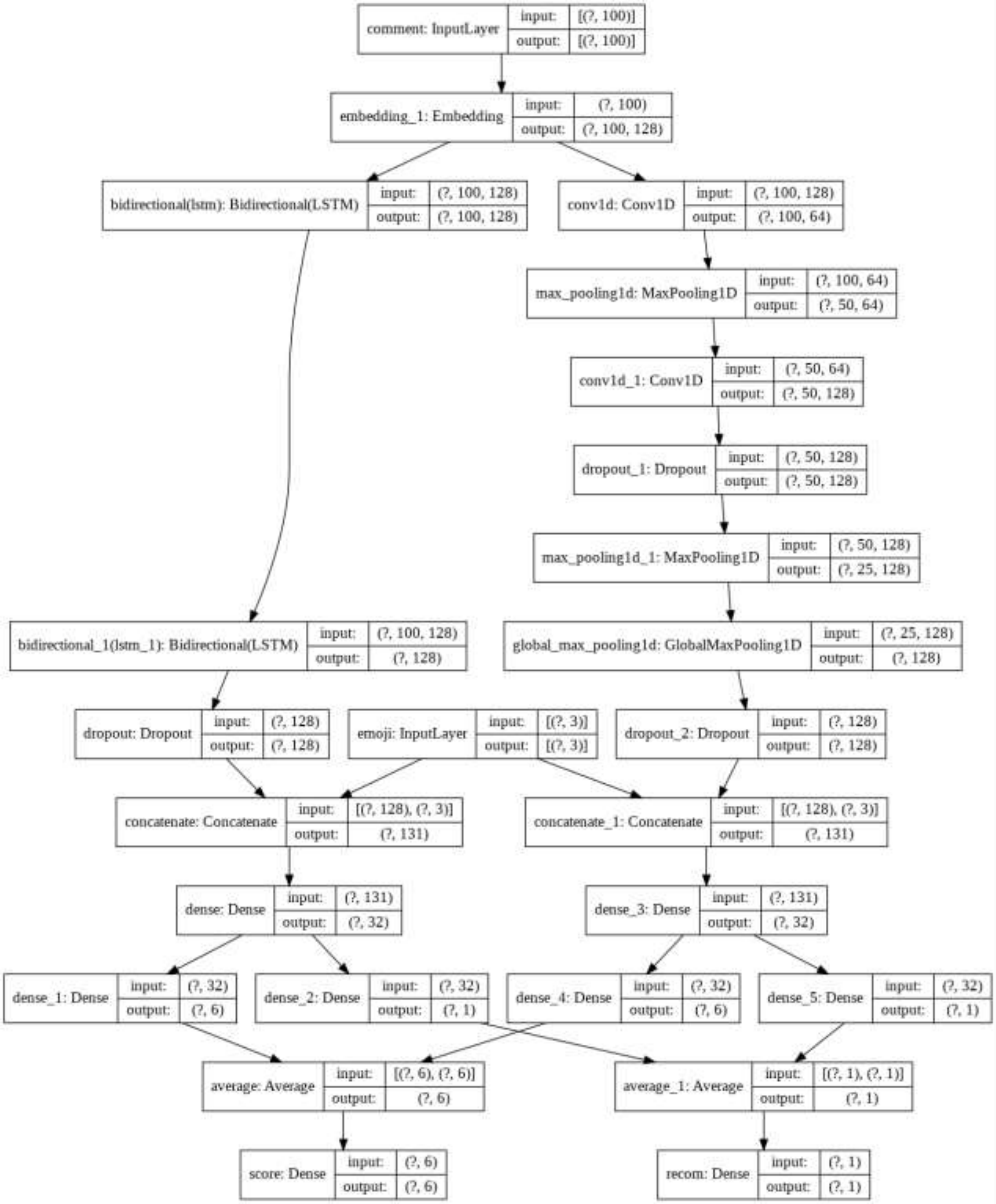
CNN and LSTM structure

Categorical cross entropy and binary cross entropy are used to calculate loss for multi classification and binary classification and total loss is weighted sum of both losses. Below fig shows two inputs, two outputs and ensemble of CNN-LSTM model in details.

Code is available here[2].

[1] https://www.kaggle.com/thomasseleck/emoji-sentiment-data

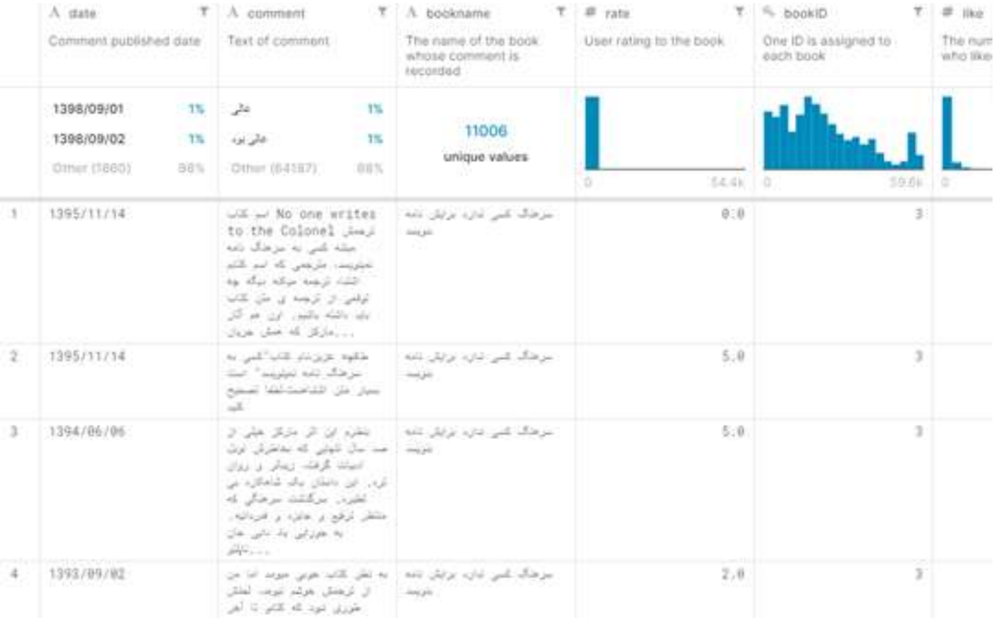
[2] https://github.com/mseslami/Sentiment-Analysis-Taghche/blob/master/taghche.ipynb


comment: InputLayer | input: [(?, 100)] | output: [(?, 100)]
embedding_1: Embedding | input: (?, 100) | output: (?, 100, 128)
bidirectional(lstm): Bidirectional(LSTM) | input: (?, 100, 128) | output: (?, 100, 128)
conv1d: Conv1D | input: (?, 100, 128) | output: (?, 100, 64)
max_pooling1d: MaxPooling1D | input: (?, 100, 64) | output: (?, 50, 64)
conv1d_1: Conv1D | input: (?, 50, 64) | output: (?, 50, 128)
dropout_1: Dropout | input: (?, 50, 128) | output: (?, 50, 128)
max_pooling1d_1: MaxPooling1D | input: (?, 50, 128) | output: (?, 25, 128)
bidirectional_1(lstm_1): Bidirectional(LSTM) | input: (?, 100, 128) | output: (?, 128)
global_max_pooling1d: GlobalMaxPooling1D | input: (?, 25, 128) | output: (?, 128)
dropout: Dropout | input: (?, 128) | output: (?, 128)
emoji: InputLayer | input: [(?, 3)] | output: [(?, 3)]
dropout_2: Dropout | input: (?, 128) | output: (?, 128)
concatenate: Concatenate | input: [(?, 128), (?, 3)] | output: (?, 131)
concatenate_1: Concatenate | input: [(?, 128), (?, 3)] | output: (?, 131)
dense: Dense | input: (?, 131) | output: (?, 32)
dense_3: Dense | input: (?, 131) | output: (?, 32)
dense_1: Dense | input: (?, 32) | output: (?, 6)
dense_2: Dense | input: (?, 32) | output: (?, 1)
dense_4: Dense | input: (?, 32) | output: (?, 6)
dense_5: Dense | input: (?, 32) | output: (?, 1)
average: Average | input: [(?, 6), (?, 6)] | output: (?, 6)
average_1: Average | input: [(?, 1), (?, 1)] | output: (?, 1)
score: Dense | input: (?, 6) | output: (?, 6)
recom: Dense | input: (?, 1) | output: (?, 1)

Model input, outputs and layers in details

4 Results

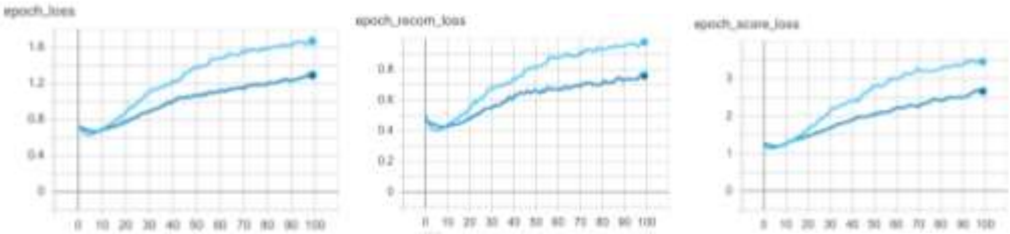
Taghche is an online e-book and audiobook store that publishes books, magazines and newspapers etc. Selected dataset is about more than 70K user's comments and 5-star rating on Taghche. User's rating is as comment's label and tells us is this book recommender or not and helps to sentiment analysis.



	A date	A comment	A bookname	# rate	bookID	# like
	Comment published date	Text of comment	The name of the book whose comment is recorded	User rating to the book	One ID is assigned to each book	The num who like
	1398/09/01 1% 1398/09/02 1% Other (1860) 98%	عالی 1% عالی بود 1% Other (64187) 98%	11006 unique values	0 54.4k	0 59.6k	0
1	1395/11/14	اسم کتاب No one writes to the Colonel ترجمش میشه کسی به سرهنگ نامه نمینویسد. مترجمی که اسم کتابم اشتباه ترجمه میکنه دیگه چه توقعی از ترجمه ی متن کتاب باید داشته باشیم. این هم اثر مارکز که عشق جریان...	سرهنگ کسی ندارد برایش نامه بنویسد	0.0	3	
2	1395/11/14	خلاصه عزیزدلم کتاب "کسی به سرهنگ نامه نمینویسد" است بسیار علی اشتباهست لطفا تصحیح کنید	سرهنگ کسی ندارد برایش نامه بنویسد	5.0	3	
3	1394/06/06	بنظرم این اثر مارکز خیلی از صد سال تنهایی که بخاطرش نوبل ادبیات گرفت، زیباتر و روان تره. این داستان یک شاهکاره بی نظیره. سرگذشت سرهنگی که منتظر ترفیع و جایزه و قدردانیه. به خوبی یاد دادی هان ...دیکتاتور	سرهنگ کسی ندارد برایش نامه بنویسد	5.0	3	
4	1393/09/02	به نظر کتاب خوبی میومد اما من از ترجمش خوشم نیومد. لحنش طوری بود که کتاب تا آخر	سرهنگ کسی ندارد برایش نامه بنویسد	2.0	3	

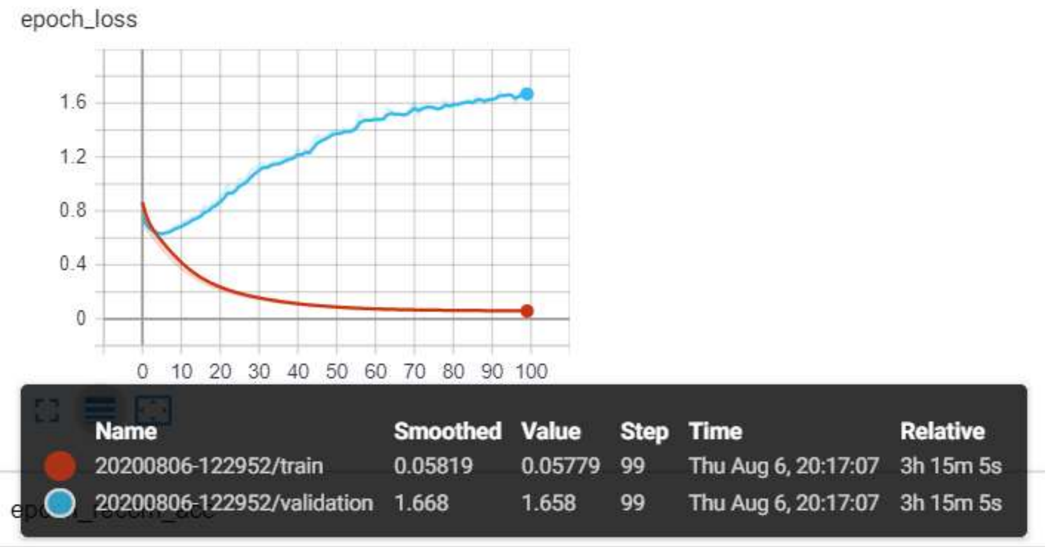
Taghche database with 6 columns on kaggle

Below figure tells us how important emoji vectors are. Different accuracy on just text data and text data concatenated with emoji-sense vector.



Light-blue: single text input; dark-blue: text input concatenated with emoji-vector

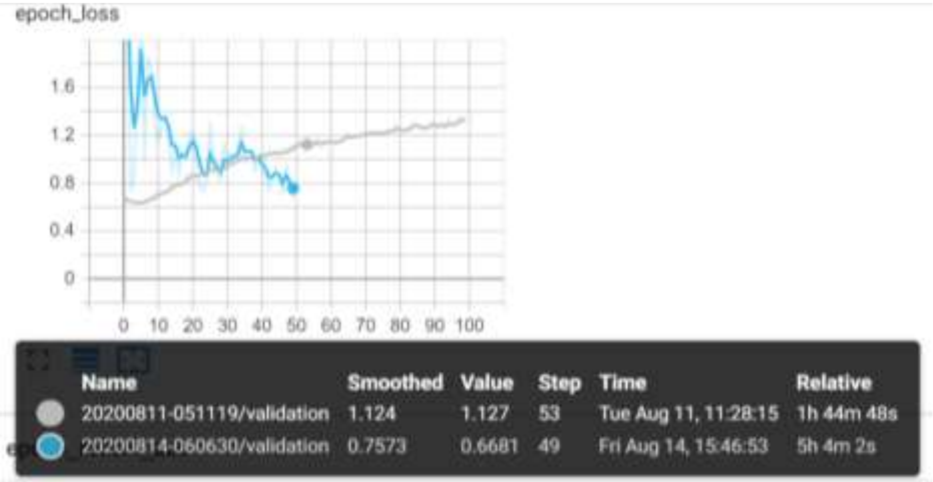
Reducing accuracy and increasing loss in training is Because of too big or too small learning rate, it will diverge and fail to find the minimum of the loss function or cannot escape from local optimum. Also larger amount of batch size reduces loss in this case. Dropout layer lead to more accuracy validation and less overfitting.



	Name	Smoothed	Value	Step	Time	Relative
●	20200806-122952/train	0.05819	0.05779	99	Thu Aug 6, 20:17:07	3h 15m 5s
●	20200806-122952/validation	1.668	1.658	99	Thu Aug 6, 20:17:07	3h 15m 5s

Overfitting and increasing validation loss

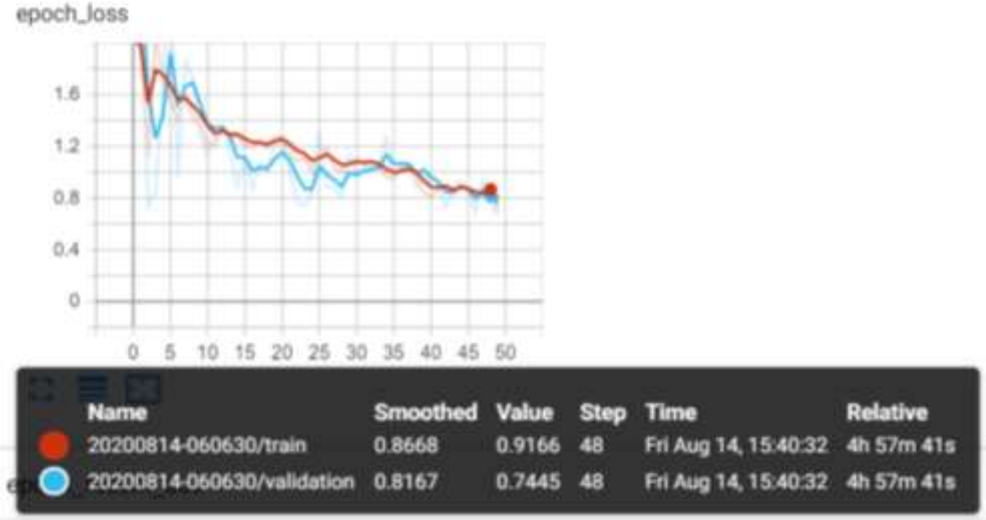
As below chart shows, by Increasing learning capacity of model, accuracy improves and loss reduces. More filters in CNN part and larger embedded vector of text etc. increase learning capacity and improved prediction and it is costly too.



	Name	Smoothed	Value	Step	Time	Relative
●	20200811-051119/validation	1.124	1.127	53	Tue Aug 11, 11:28:15	1h 44m 48s
●	20200814-060630/validation	0.7573	0.6681	49	Fri Aug 14, 15:46:53	5h 4m 2s

Blue-line: The larger model with less loss

Final result with fine accuracy with reducing loss and increasing accuracy in 5 epochs without overfitting:



	Name	Smoothed	Value	Step	Time	Relative
●	20200814-060630/train	0.8668	0.9166	48	Fri Aug 14, 15:40:32	4h 57m 41s
●	20200814-060630/validation	0.8167	0.7445	48	Fri Aug 14, 15:40:32	4h 57m 41s

final result: total loss in 50 epochs

5 Discussion

Sentence is not just a single vector. It contains some meaningful tokens which are related to each other. In this task we access a huge dataset with vary inputs in context and size thus we are ready to train a deep neural network. In sentimental analysis, as a NLP classification task, recurrent neural networks are a candidate because RNNs can remember. Relation and association of some words with same positive or negative sense lead to remember and learn sentence's sense. In other hand CNNs learn local features and increasing number of CNN layers in a stack of CNNs lead to gather more global features. These two models are proposed with different point of view to process comments. Researches demonstrate that ensemble multimodels increase accuracy and robustness and model improves.

In this report emojis play an important role on text sense. An emoji-vector concatenate to embedded and processed text and the concatenated vector fed to dense layers. Results show The emoji-vector directly influence output.

Results show that using word embedding can effectively increase performance. Depending on different social medias, performance might vary. In Instagram comments or twitter more emojis are used and it is easier to be classified with higher accuracy. In this project, noise comments are significant. More training data leads to better performance.

State of the art of text processing is attention base approaches. Transformers based on self-attention for long text inputs are good choice. Furthermore, pretrained models can compete with state of the art of same task. Theoretically using pretrained model definitely improves model's performance.

6 References

[1] P. D. Turney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews," *Association for Computational Linguistics ,* pp. 417-424, 2002.

[2] S. Baccianella, A. Esuli and F. Sebastiani, SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining, Valletta, Malta: European Language Resources Association, 2010.

[3] H. Kanayama and T. Nasukawa, "Fully Automatic Lexicon Expansion for Domain-oriented Sentiment Analysis," *Association for Computational Linguistics,* p. 355–363, 2006.

[4] M. Taboada, J. Brooke, M. Tofiloski, K. Voll and M. Stede, "Lexicon-Based Methods for Sentiment Analysis," *Computational Linguistics,* vol. 37, no. 2, pp. 267-307, 2011.

[5] T. Mullen and N. Collier, "Sentiment Analysis using Support Vector Machines with Diverse Information Sources," *Association for Computational Linguistics,* pp. 412-418, 2004.

[6] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng and C. Potts, "Learning Word Vectors for Sentiment Analysis," *Association for Computational Linguistics,* p. 142–150, 2011.

[7] S. Chao, W. Xiao-Kang, C. Peng-fei, W. Jian-qiang and L. Lin, "SACPC: A framework based on probabilistic linguistic terms for short text sentiment analysis," *Knowledge-Based Systems,* vol. 194, p. 105572, 2020.

[8] N. Usman, R. Imran, M. Katarzyna and I. Muhammad, "Transformer based Deep Intelligent Contextual Embedding for Twitter sentiment analysis," *Future Generation Computer Systems Volume,* vol. 113, pp. 58-69, 2020.

[9] Y. Chen, J. Yuan and J. Luo, "Twitter Sentiment Analysis via Bi-sense Emoji Embedding and Attention-based LSTM," *Association for Computing Machinery,* pp. 117-125, 2018.