From Writings to Writers

Text Classification on Blog Authorship Corpus
Shangyun Lv
School of Information, University of California, Berkeley

Abstract

In this course project, I conducted a text classification task using neural network model on the Blog Authorship Corpus and author profile related label (mainly gender) to see if the model could learn the difference on different writers' writing style, content, and other features. Two model structures are used in the project, one use global average pooling and fully connected layers while the other use convolutional layer and global max pooling layer. Pre-trained and self-trained word embedding are experimented in both model structures to compare the classification performance.

Introduction

In his insightful quote -"The writing mirrors the writer." -- Shi Su, an ancient Chinese writer, suggested the existence of certain knowledge embedded in languages and how people utilize it. However, he didn't elaborate on what the knowledge is. Fast forward 1000 years to the age of the internet and artificial intelligence, I believe we now possess enough data and powerful methods and tools to re-visit Su's insight to see if we could clear a bit the mist of whether different people use language differently. If so, can we tell?

The goal of this project is to address the above question by trying to a few text classification algorithms to predict the profile of writers such as gender and the industry that writer works in. Acknowledging the variety of contexts that languages are used in which different rules and styles are applied, this project by no means intends to address all kinds of language use. According to past studies (Berryman-Fink and Wilcox 1983; Simkins-Bullock andWildman 1991) , one should expect no difference between author with different profiles (mainly gender) in more formal contexts. Therefore, I select one of the groups of text with the least form restriction and the most free style - blog post. Such text selection is on the assumption that texts with the most variations may reveal the most information about the writers which helps the algorithms on the classification task.

The specialty about this project compared with other natural language processing tasks like sentiment analysis, translation, and summarization is that sensing sentiment in language, translation and summarization are natural skills of people. There are -- though may not be explicit -- ways and standards to achieve these tasks. However, describing the authors based on their writings is not a common skill. This may suggest some risk in the project as to whether the knowledge is too implicit to learn by the algorithm, this could also mean certain challenge on improving the algorithm for better classification performance.

Background

Past studies have attempted similar tasks using different methods and data. Shlomo Argamon et.al demonstrated the writing difference between male and female authors by comparing lexical and syntactic features such as the frequency of certain types of words using British National Corpus (BNC). One of the discoveries from their study is that female writers use pronouns significantly more than male authors do. Jonathan Schler et.al use Multi-Class Real Winnow (MCRW) to train a model on predicting the gender of the author using the Blog Authorship Corpus(the same dataset used in this project) and is able to reach 80% accuracy. Both of these studies rely on features generated by the researcher using domain knowledge. Shlomo uses word frequency from a grammar perspective and Jonathan uses word frequency from style and content perspective. In this project, I believe there might be implicit features in the writing beyond grammar, style and content. Neural networks might be a good method to learn such implicit features and may be able to achieve similar if not better prediction performance.

Methods

Data and Wrangling

I use the Blog Authorship Corpus for this project which incorporates a total of 681,288 blog posts (over 140 million words) from 19,320 bloggers gathered from blogger.com in August 2004 - or approximately 35 posts and 7250 words per person. The corpus also provide four pieces of information about the authors - gender, age, industry, and astrological sign - as the labels for the classification.

The length of the posts formed a very skewed distribution as in Figure 1.

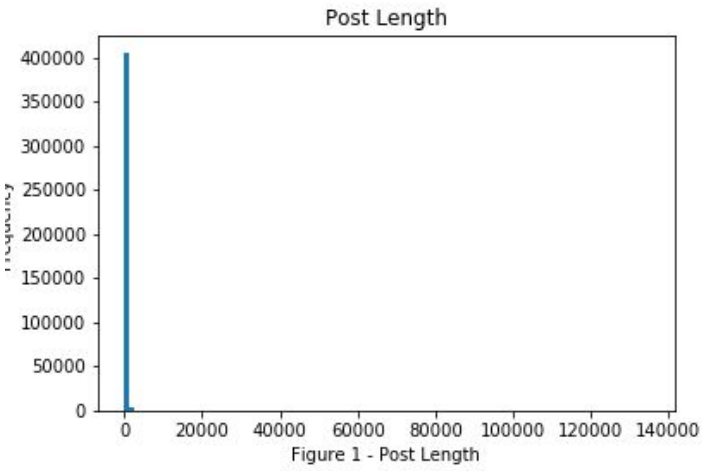


Figure 1 - Post Length

	Post Length
mean	211.19
std	470.64
min	0
25%	38
50%	118
75%	269
90%	493
max	134782

As in the table, 90% of the posts are within the length of 500 words. I took a look at those extremely long posts and many of them look like Figure 2 and Figure 3.

```
'BLUE M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A
N B L U E M A N B L U E M A NB L U E M A N B L U E M A N B L U E M A N B L U E
M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A N B
L U E M A N B L UE MANB L U E M A N B L U E M A N BL U E M A N B L U E M A N B
L U E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M
A N B L U E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A N B L
U E M A N B L U E M A N B L U E M A NB LUEM A N B L U E M A N B L U E M A N B L
U E M AN B L U E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A N
B L U E M A N B L U E M A N B L U E M A N BL U E M A N B L U E M A N B L U E M
A N B L U E M A N B L U E M A N B L U E M A N B LUEM ANBL U E M A N B L U E M A
N B L U E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A N B L U
E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M ANB L
U E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A N BL UEMAN B L
U E M A N B L U E M A N B L U E M A N B L U EM A N B L U E M A N B L U E M A N
BLU EM ANBLU E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A NB L U
E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A N B L UEMA NBLU E
M A N B L U E M A N B L U E M A N B L U E M A N BL U E M A N B L U E M A N B L
U E MANB L UEMAN BLU EM A N B L U E M A N BL U E M A N B L U E M A N B LU E M A N
B L U E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A N B LU EMAN
B L U E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A N B L U E
M ANBLUEMANBLUEMANBL U E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A
N B L U E M A N B L U E M A N B L U E M A N B L U E M A N B L U E MAN BLUE M A
N B L U E M A N B L U E M A NB L U E M A N B L U E M A N B L U E M A N B LUEMANBL
UEMANBLUEMA N B L U E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A
N B L U E M A N B L U E M A N B L U E M AN B L U E M A N B LUE MANB L U E M A N
B L U E M A N B L U E MA N B L U E M A N B L U E M A N B L U E M ANBLUEMANBLUEMANBLU
E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A N
B L U E M A N B L U E M A N B L U E M A N B L U E M A NB L U E M A N B L U E M
A NB L U E M A N B L U E M A N B L U E M A N B L U E M A N B L UEMANBLUEMANB LU E M
A N B L U E M A N B L U E M A N B L U E M A N B L U E M A N B L U E M A N B L
U E M A N B L U E M A N B L U E M A N B L U E M A N B L U EM A N B L U E M A N
B L U E M A N B L U E M A NBL U E M AN B L U E M A N B L U E M A NBLUEMANBLUEM AN B
```

Figure 2 - one example of extremely long post

```
"and here's ur return gift :- I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU
I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE
YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I L
OVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU
I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE
YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I L
OVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU
I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE
YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I L
OVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU
I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE
YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I L
OVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU
I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE
YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I L
OVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU
I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE
YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I L
OVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU
I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE
YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I L
OVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU
I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE
YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I L
OVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU
I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE
YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I L
OVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU
I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE
YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I L
OVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU
I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE YOU I LOVE
```

Figure 3 - another example of extremely long post

It is fair to say that most of the long posts do not contain much more information than the short post. Therefore, I decided to truncate the long posts. If a post is longer than 500 words (close to 90% quantile post length), I will split the post into two posts, one using the first 500 words and the other using the last 500 words. Since this operation only add less than 10% of the data with

specific labels into the dataset, it doesn't not heavily distort the label distribution. I will also use class weight argument in the model to compensate for the imbalance.

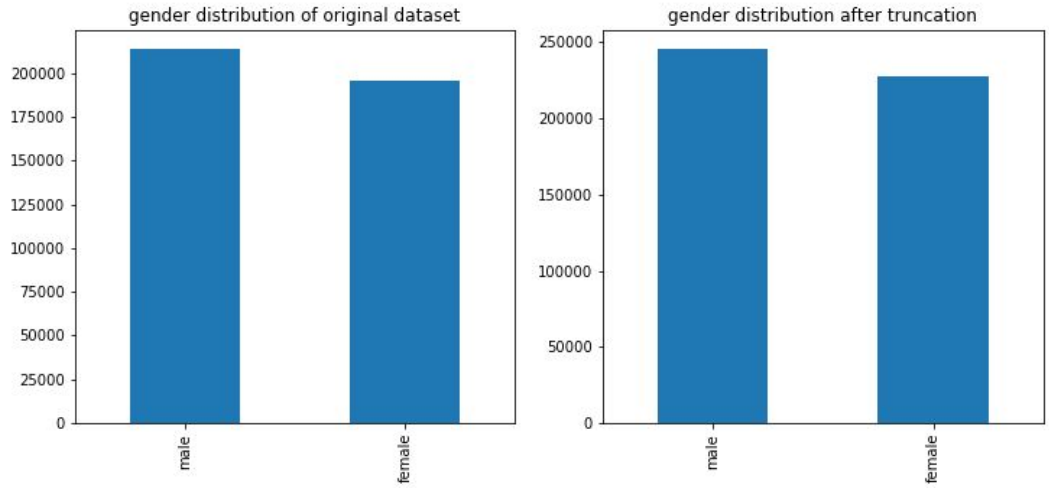


Figure 4 - label distribution before and after truncation

Baseline Model

Inspired by the work of Shlomo and Jonathan in which different word frequencies used as features for the classification, I choose Bag of Words and logistic regression as my baseline model which also relies on word count as features. I also choose logistic regression for the classification with 5 fold cross validation. The model achieves 65% accuracy on the test data. My objective is to use neural network to achieve better accuracy.

Modeling

I applied two different neural network structures to the classification task in the project. The first one is inspired by the tensorflow documentation, which is constructed first with a word embedding layer followed by a global average pooling layer which then followed by a few fully connected layers. The final output layer will be a softmax layer.

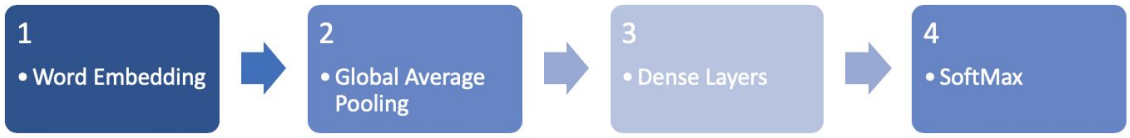


Figure 5 - Model 1 Structure

The other structure is the application of the Convolutional Neural Networks for Sentence Classification by Yoon Kim.

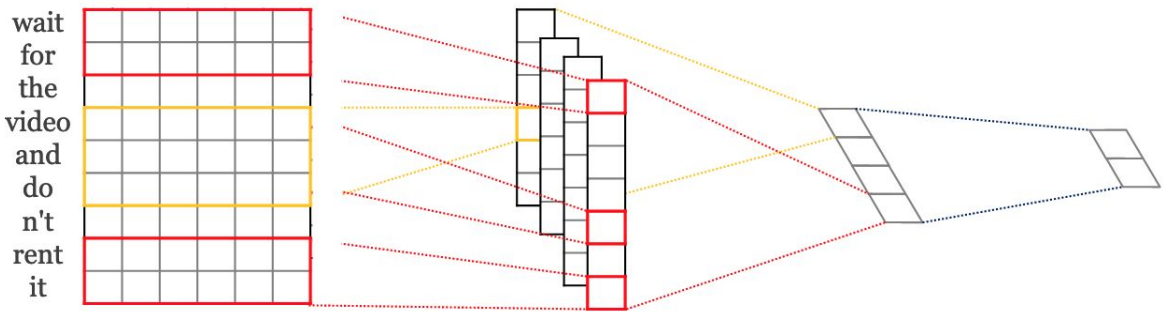
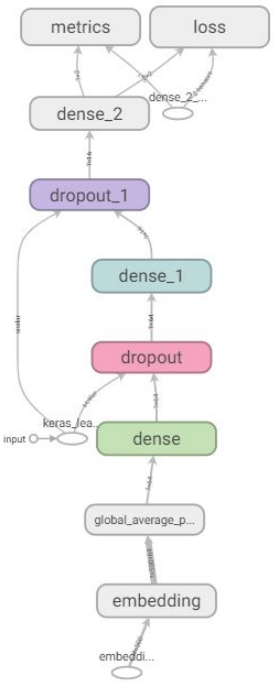


Figure 6 - Model 2 Structure

One implementation option this project will experiment is the performance difference between Pre-trained word embedding vs. Self-train word embedding. The project will experiment GloVe word embedding(6B tokens, 400K vocab, uncased, 100d vectors) and self-trained embedding in the embedding layer of the model to see which achieve better classification result.

Given the limitation of time and resources, Not all hyper-parameters are tuned in the modeling training process. Below are a few parameters are set fixed for all models:

- The tokenizer of the text is set to use the most frequent 10000 words for the modeling;
- Only GloVe embedding (6B tokens, 400K vocab, uncased, 100d vectors) is used as the pre-trained embedding, other Variations of GloVe embeddings or Word2Vec embedding is not used;
- The activation function of all hidden layers are set to RELU;
- The optimization function of all models is set to use ADAM with beta_1 as 0.9, beta_2 as 0.999 and with no learning rate decay;
- 10% of the validation split is used during training for all models;
- All models run through 20 epochs during training



For hyper-parameters to be optimized, grid search is used for tuning in all the models in the project, sightly different hyper-parameters are tuned for different models based on the model structure.

Model 1

Model 1 is constructed with layers in the following sequence:

1. Word embedding layer
2. Global average pooling 1D
3. Dense layer with dropout
4. Dense layer with dropout

5. Dense layer for Output

Figure 7 to the right indicates the model structure.

Model with more and less dense layers are experimented with no better performance with the two dense hidden layers.

Hyper-parameters tuned during training are:

Hyper-parameter	Value 1	Value 2	Value 3
Learning Rate	0.0001	0.001	0.1
Dropout Rate	0.1	0.2	0.3
No. of Unit in the last hidden layer	16	32	
Mini-batch size	128	256	512

Result and Discussion

For Model 1 with pre-trained word embedding(GloVe) with tuned hyper-parameters, the best accuracy on the validation set is 0.642. The best set of hyper-parameter is:

- Learning Rate : 0.001
- Dropout Rate : 0.1
- No. of Unit in the last hidden layer : 16
- Mini-batch size : 128

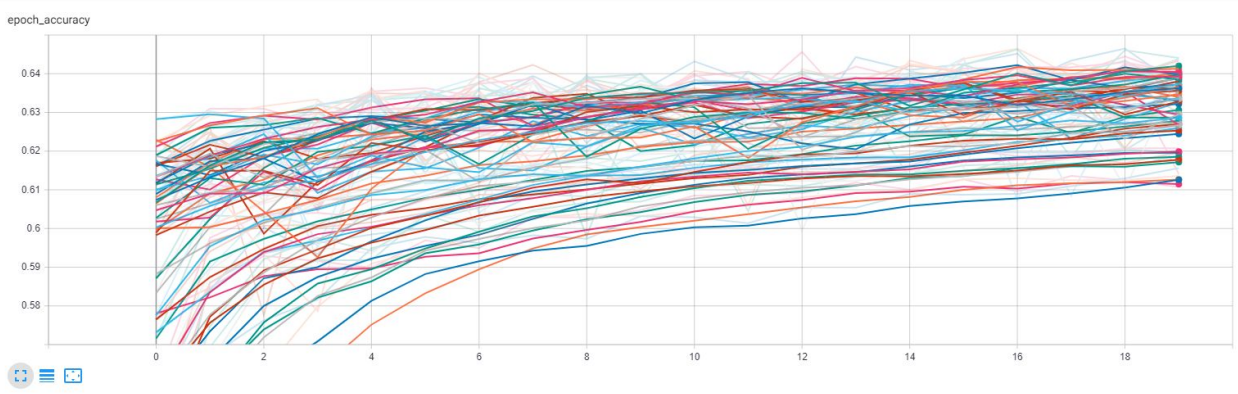


Figure 8 - Accuracy on Validation Set of Model 1 with GloVe Embedding

The best accuracy on the test set is 0.643. Accordingly, the best set of hyper-parameter is:

- Learning Rate : 0.01
- Dropout Rate : 0.2
- No. of Unit in the last hidden layer : 16
- Mini-batch size : 128

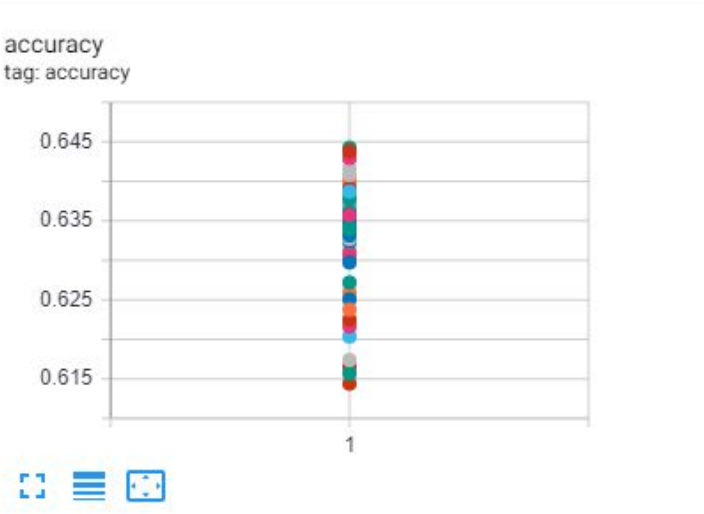


Figure 9 - Accuracy on Test Set of Model 1 with GloVe Embedding

For Model 1 with the self-trained word embedding with tuned hyper-parameters, the best accuracy on the validation set is 0.683. The best set of hyper-parameter is:

- Learning Rate : 0.0001
- Dropout Rate : 0.1
- No. of Unit in the last hidden layer : 32
- Mini-batch size : 128

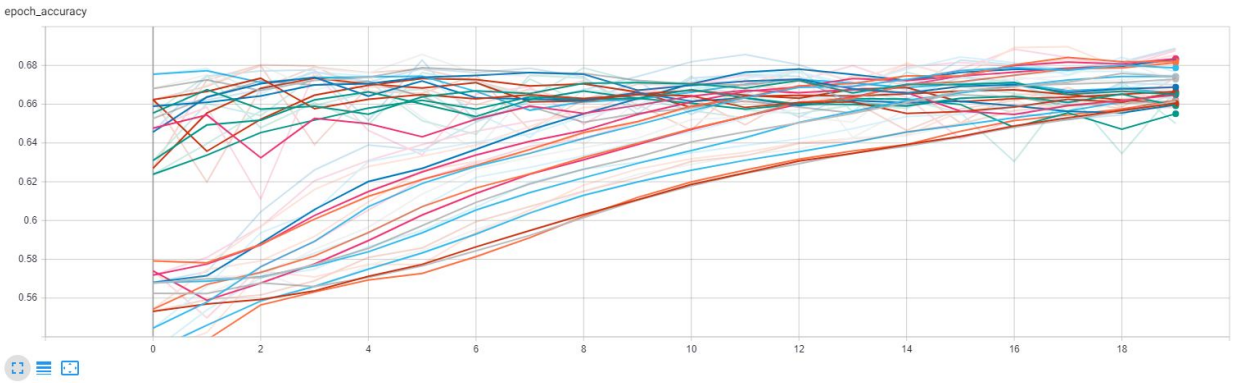


Figure 10 - Accuracy on Validation Set of Model 1 with Self-trained Embedding

The best accuracy on the test set is 0.695. Accordingly, the best set of hyper-parameter is:

- Learning Rate : 0.0001
- Dropout Rate : 0.1
- No. of Unit in the last hidden layer : 32

- Mini-batch size : 512

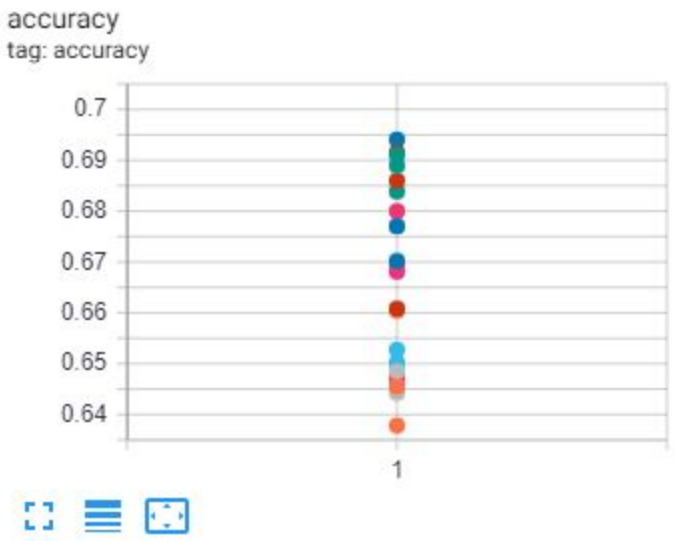


Figure 11 - Accuracy on Test Set of Model 1 with Self-trained Embedding

Comparing the best performance of the two variations(Pre-trained vs. Self-trained word embedding) of Model 1, the self-trained word embedding version works better than the pre-trained one. This aligns with the expectation in that the corpus that the corpus that GloVe embedding is trained on is different from the blog authorship corpus in this project which contains very informal expression or even expression with errors. Take below blog posts as an example, “kabloosh” and “=P” doesn’t exist in GloVe but are quite popular in online blogs. Also GloVe may not interpret “Sorry ‘bout that” correctly due to the abbreviation.

“ Err... i think i get his point. Sorry 'bout that... Well, maybe for you it won't be a critical status to be "psychedelic", but when you're in there knowing nothing about what's going to happen next, then "kabloosh..." =P But the reason i smoke? I really don't know. But now all i know is that i have this one "judgement stick" left in my W-Lights' case. If i take that one, i won't smoke again... darn.”

Model 2

Model 2 is constructed based on the sentence classification paper by Yoon Kim. The model is constructed with an embedding layer followed by four parallel convolutional layers and global maxing pooling layers. The output of the global max pooling is concatenated together and feed into a fully connected layer and finally the output layer.

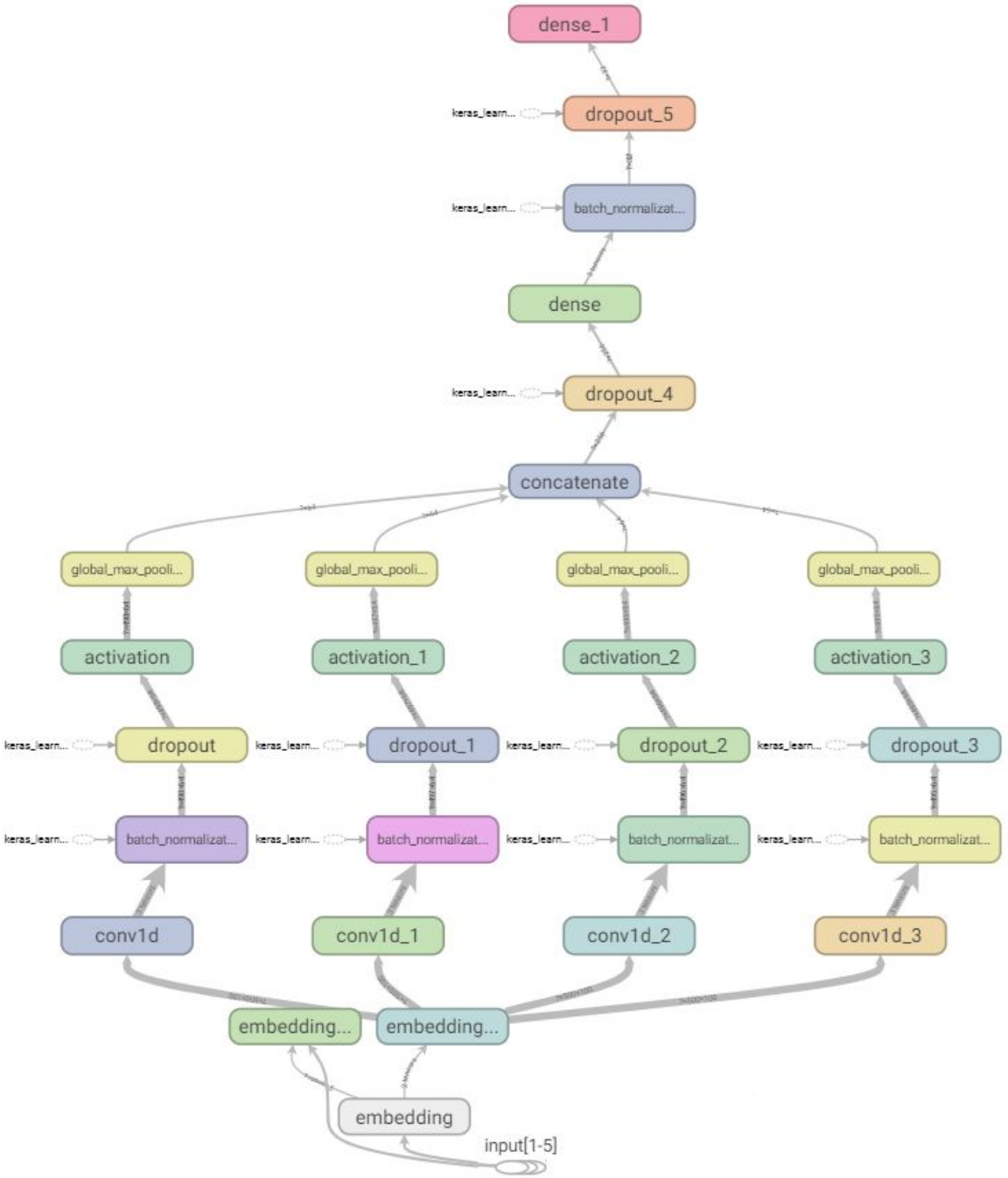


Figure 12 - Model 2 Structure

Hyper-parameters tuned during training are:

Hyper-parameter	Value 1	Value 2	Value 3
Learning Rate	0.0001	0.001	0.1

Dropout Rate	0.1	0.2	0.3
No. of Filter	64	128	
Mini-batch size	128	256	512

Result and Discussion

For Model 2 with pre-trained word embedding(GloVe) with tuned hyper-parameters, the best accuracy on the validation set is 0.646. The best set of hyper-parameter is:

- Learning Rate : 0.001
- Dropout Rate : 0.1
- No. of Filter : 64
- Mini-batch size : 256

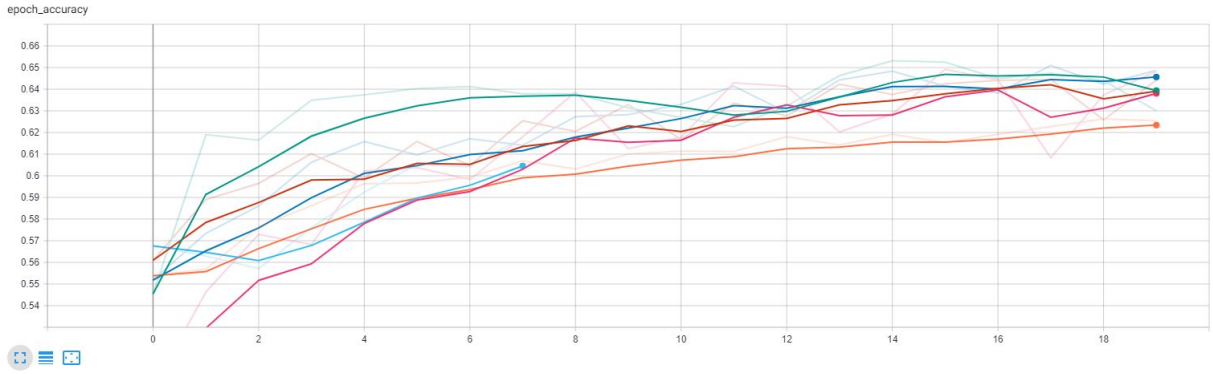


Figure 13 - Accuracy on Validation Set of Model 2 with GloVe Embedding
(a few lines are removed to avoid clustering)

The best accuracy on the test set is 0.646. Accordingly, the best set of hyper-parameter is:

- Learning Rate : 0.01
- Dropout Rate : 0.2
- No. of Filters : 64
- Mini-batch size : 128

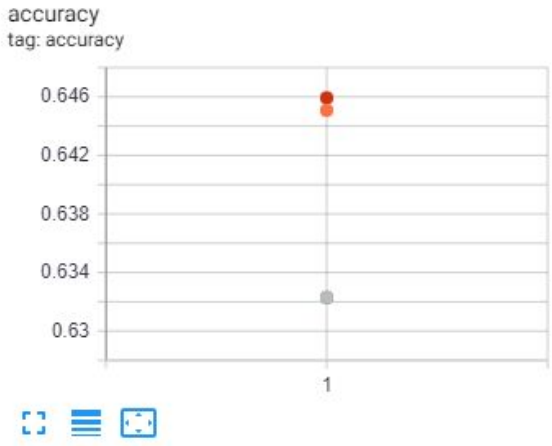


Figure 14 - Accuracy on Test Set of Model 2 with GloVe Embedding
(a few points are removed to avoid clustering)

For Model 2 with the self-trained word embedding with tuned hyper-parameters, the best accuracy on the validation set is 0.679. The best set of hyper-parameter is:

- Learning Rate : 0.0001
- Dropout Rate : 0.1
- Mini-batch size : 256

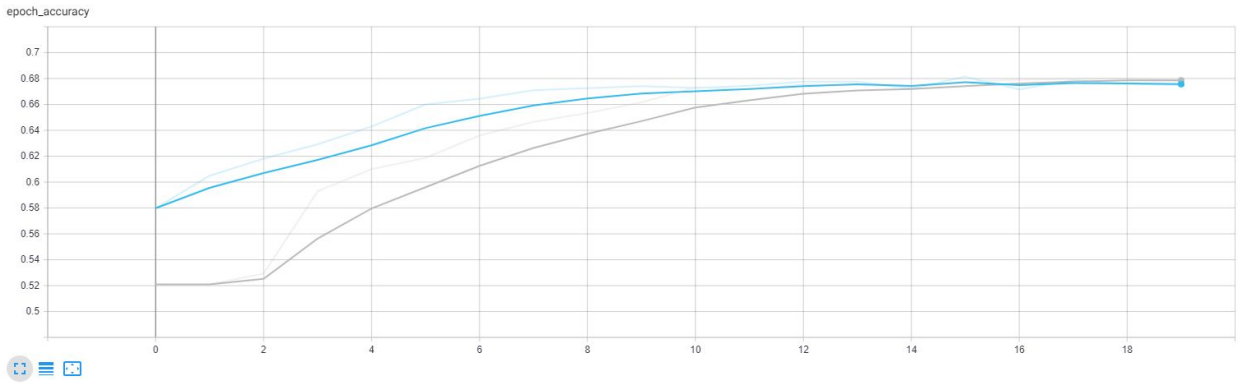


Figure 15 - Accuracy on Validation Set of Model 2 with Self-trained Embedding
(a few lines are removed to avoid clustering)

The best accuracy on the test set is 0.659. Accordingly, the best set of hyper-parameter is:

- Learning Rate : 0.0001
- Dropout Rate : 0.1
- Mini-batch size : 128

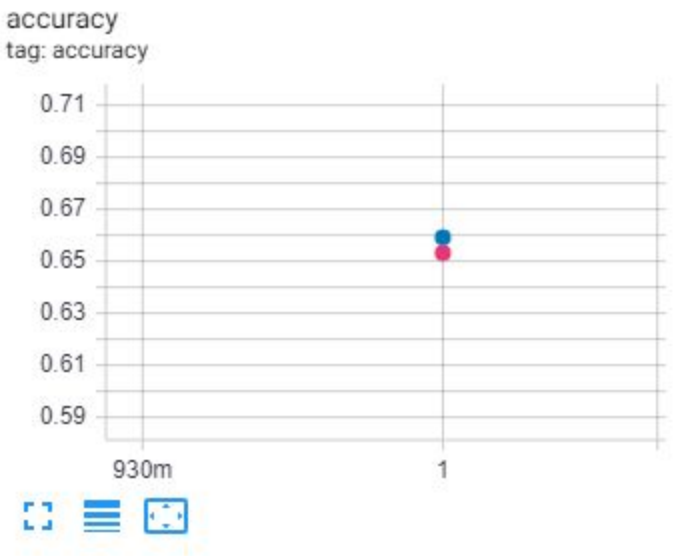


Figure 16 - Accuracy on Test Set of Model 2 with Self-trained Embedding

(a few lines are removed to avoid clustering)

Comparing the best performance of the two variations(Pre-trained vs. Self-trained word embedding) of Model 2, we can reach similar conclusion as Model 1. The self-trained word embedding performed better than pre-trained word embedding version.

Conclusion

By applying different structures of neural network model (model 1 and model 2) on the same text classification, we are able to compare the performance of different implementations. We found that the self-trained word embedding perform better than the pre-trained word embedding in both model structures. Compared with the baseline model which achieve 65% accuracy on the test set, the self-trained embedding for both model structures achieve better classification performance than the baseline model whereas the pre-trained word embedding is not able to achieve 65% accuracy of the baseline model.

Model	Accuracy on Test Set with the best h-params
Baseline	65%
Model 1 with Pre-Trained Word Embedding	64.3%
Model 1 with Self-Trained Word Embedding	69.5%
Model 2 with Pre-Trained Word Embedding	64.6%
Model 2 with Self-Trained Word Embedding	65.9%

Code: https://github.com/sylvao08/W266-2019Fall-Writing-to-Writer

References

Background and Domain

[1] Berryman-Fink, C. L. and Wilcox, T. R. (1983). A multivariate investigation of perceptual attributions concerning gender appropriateness in language. Sex Roles 9: 663–681.

[2] Simkins-Bullock, J. A. and Wildman, B. G. (1991). An investigation into the relationship between gender and language. Sex Roles 24: 149–160.

[3] Argamon, S., M. Koppel , J. Fine and A. Shimoni (2003), Gender, Genre, and Writing Style in Formal Written Texts, Text , 23(3), 2003
http://kanagawa.lti.cs.cmu.edu/amls09/sites/default/files/argamon03.pdf

[4] J.Schler, M. Koppel, S. Argamon and J. Pennebaker (2006). Effects of Age and Gender on Blogging in Proceedings of 2006 AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs.
http://u.cs.biu.ac.il/~schlerj/schler_springsymp06.pdf

Method

[5] Text classification with preprocessed text: Movie reviews
https://www.tensorflow.org/tutorials/keras/text_classification
[6] Y.Kim (2014). Convolutional Neural Networks for Sentence Classification in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751
https://www.aclweb.org/anthology/D14-1181.pdf