HW1

xy990

October 2018

1 Hyper-parameters Tuning

1.1 Tokenize

For the parameter tokenizer tuning, I used 3 techniques, one is only remove punctuation, another is only remove stop words, another is stemming, besides, I also combine three methods together remove punctuation, stop words and stemming. Compared with validation accuracy, only the methods which only remove stop words performs best. Please see results in figure 1 and 2 .

1.2 N-grams

For the parameter N-grams tuning, I choose N-grams is 1, 2, 3, 4. Compared with the validation accuracy, unigram performs best.Please see results in figure 3 and 4 .

1.3 Optimizer

For the optimizer I choose to compare the performance of SGD and Adam, it is a little difficult to compare them because they have different performance with different magnitude of learning rate. Therefore, for SGD with momentum of 0.9, I choose the model with learning rate 0.01, 0.1 and 1. For Adam, I choose the model with learning rate 0.001, 0.01, 0.1. Due to early stopping, model with Adam and learning rate 0.01 early stops within 5 epochs, so I do not plot this result in the learning curve figures. For other choice, they do not meet early stopping, the results show Adam and learning rate 0.001 performs best, and

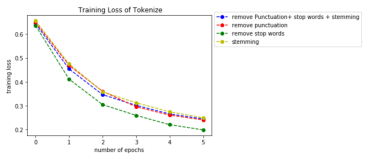

Figure 1: training loss of tokenizer

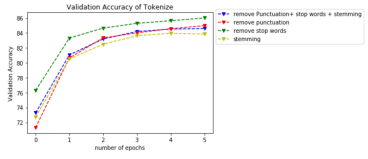

Figure 2: validation accuracy of tokenizer

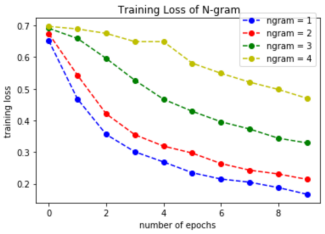

Figure 3: training loss of N-grams

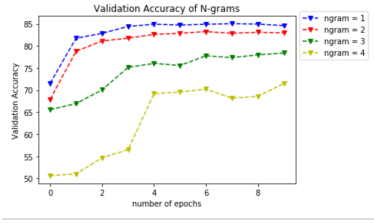

Figure 4: validation accuracy of N-grams

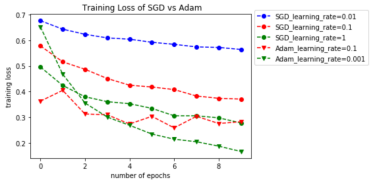

Figure 5: training loss of Optimizer(SGD vs Adam)

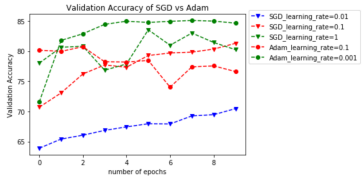

Figure 6: validation accuracy of Optimizer(SGD vs Adam)

SGD and learning rate 1 performs good, but it fluctuate in validation accuracy due to large learning rate.Please see results in figure 5 and 6 .

1.4 Learning Rate

For parameter learning rate, I choose SGD with momentum 0.9 as optimizer and learning rate 0.001, 0.01, 0.1, 1. Due to the very bad performance of learning rate 0.001 for SGD in 10 epochs, I do not plot and show it in the table. Compared with the validation accuracy, model with SGD and learning rate 1 performs best, but the validation accuracy fluctuates and does not converge. Please see results in figure 7 and 8 .

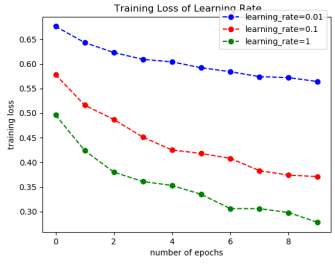

Figure 7: training loss of Learning Rate

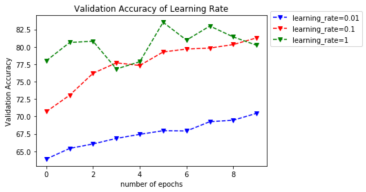

Figure 8: validation accuracy of Learning Rate

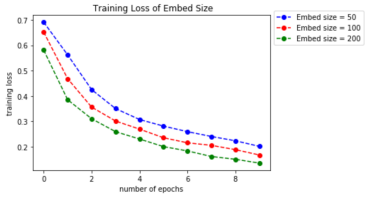

Figure 9: training loss of Embed Size

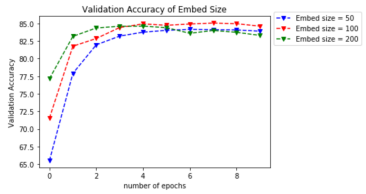

Figure 10: validation accuracy of Embed Size

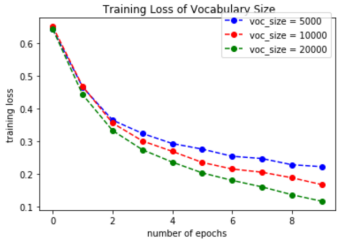

Figure 11: training loss of vocabulary size

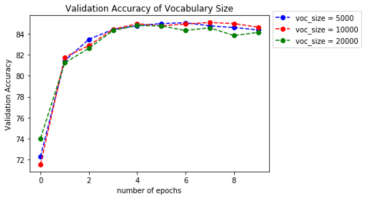

Figure 12: validation accuracy of vocabulary size

1.5 Embedding Size

For embedding size, I choose 50, 100 and 200. Compared with the validation accuracy, embedding size of 100 performs best. Please see results in figure 9 and 10 .

1.6 Vocabulary Size

For vocabulary size, I choose 5000, 10000 and 20000. I expect vocabulary size larger, the performance should be better. However, compared with the validation accuracy, vocabulary size of 5000 performs slightly better than others. Please see results in figure 11 and 12 .

1.7 Table of Training Loss and Validation Accuracy for All Parameters Tuning

For the results of all parameters tuning, please see table 13 and 14 .

2 Best Model and Error Analysis

After tuning the parameters, I choose remove stop words for tokenizer, 1 for N-grams, 5000 for vocabulary size, Adam for optimizer, 100 for embedding size and 0.001 for learning rate, the model reaches Val Acc 84.54 and Test Acc 85.8.

Tokenize	N-gram	Vocab-size	Optimizer	embed size	Learning Rate	l1	l2	l3	l4	l5	l6	l7	l8	l9	l10
remove Punc, stop words + stem	1	10000	adam	100	0.001	0.644	0.454	0.346	0.301	0.265	0.244				
remove punctuation	1	10000	adam	100	0.001	0.65	0.469	0.36	0.295	0.26	0.24				
remove stop words	1	10000	adam	100	0.001	0.636	0.411	0.304	0.259	0.22	0.198				
stemming	1	10000	adam	100	0.001	0.656	0.475	0.358	0.312	0.274	0.249				
remove punctuation	1	10000	SGD	100	0.1	0.578	0.516	0.487	0.451	0.425	0.418	0.408	0.383	0.374	0.371
remove punctuation	1	10000	SGD	100	1	0.496	0.424	0.38	0.361	0.353	0.335	0.306	0.306	0.298	0.278
remove punctuation	1	10000	SGD	100	0.01	0.676	0.643	0.623	0.609	0.604	0.592	0.584	0.574	0.572	0.564
remove punctuation	1	10000	Adam	100	0.1	0.363	0.405	0.313	0.31	0.274	0.304	0.259	0.304	0.276	0.283
remove punctuation	1	10000	Adam	100	0.01	0.318	0.318	0.182	0.143	0.112					
remove punctuation	1	10000	Adam	100	0.001	0.652	0.468	0.356	0.301	0.269	0.235	0.215	0.205	0.188	0.167
remove punctuation	2	10000	Adam	100	0.001	0.674	0.543	0.421	0.355	0.319	0.297	0.264	0.243	0.231	0.214
remove punctuation	3	10000	Adam	100	0.001	0.691	0.659	0.596	0.527	0.467	0.429	0.396	0.373	0.344	0.329
remove punctuation	4	10000	Adam	100	0.001	0.697	0.689	0.675	0.649	0.649	0.581	0.549	0.521	0.498	0.47
remove punctuation	1	10000	Adam	50	0.001	0.691	0.562	0.425	0.35	0.307	0.281	0.259	0.24	0.223	0.201
remove punctuation	1	10000	Adam	200	0.001	0.581	0.385	0.31	0.259	0.23	0.2	0.183	0.161	0.15	0.135
remove punctuation	1	20000	Adam	100	0.001	0.644	0.443	0.333	0.274	0.236	0.203	0.18	0.16	0.136	0.116
remove punctuation	1	5000	Adam	100	0.001	0.644	0.466	0.365	0.324	0.293	0.276	0.254	0.247	0.228	0.222

Figure 13: train loss table for all parameters tuning

Tokenize	N-gram	Vocab-size	Optimizer	embed size	Learning Rate	acc1	acc2	acc3	acc6	acc5	acc8	acc7	acc8	acc9	acc10	Testacc	Valacc
remove Punc, stop words + stem	1	10000	adam	100	0.001	75.28	81.06	83.2	84.2	84.52	84.6					85.246	84.58
remove punctuation	1	10000	adam	100	0.001	71.3	80.66	83.34	84.02	84.58	84.96					85.476	84.73
remove stop words	1	10000	adam	100	0.001	76.3	83.32	84.66	85.3	85.64	86.02					86.6	85.87
stemming	1	10000	adam	100	0.001	72.74	80.5	82.5	83.64	83.96	83.86					84.432	83.7
remove punctuation	1	10000	SGD	100	0.1	70.7	73.06	76.2	77.7	77.32	79.26	79.68	79.82	80.32	81.28	81.65	81.8
remove punctuation	1	10000	SGD	100	1	78	80.62	80.8	76.82	77.86	83.52	80.96	82.98	81.42	80.22	83.016	83.2
remove punctuation	1	10000	SGD	100	0.01	63.9	65.42	66.98	66.84	67.44	67.96	67.92	69.26	69.46	70.66	70.8	71.1
remove punctuation	1	10000	Adam	100	0.1	80.14	79.94	80.72	78.22	78.18	78.44	74.02	77.38	77.54	76.6	79.896	79.02
remove punctuation	1	10000	Adam	100	0.01	84.84	81.84	82.34	81.84	80.74						82.14	80.74
remove punctuation	1	10000	Adam	100	0.001	71.54	81.76	82.86	84.42	84.04	84.74	84.92	85.06	84.96	84.62	85.184	84.48
remove punctuation	2	10000	Adam	100	0.001	67.92	78.8	81.14	81.76	82.62	82.88	83.26	82.88	83.1	82.98	83.236	82.46
remove punctuation	3	10000	Adam	100	0.001	65.56	66.94	70.04	75.14	76.06	75.58	77.78	77.36	77.58	76.4	78.284	77.43
remove punctuation	4	10000	Adam	100	0.001	50.6	51	54.68	56.48	60.2	60.58	70.2	68.2	68.56	71.52	72.588	71.66
remove punctuation	1	10000	Adam	50	0.001	65.52	77.9	81.96	83.2	83.76	84.02	84.16	84.12	84.08	83.9	85.112	83.84
remove punctuation	1	10000	Adam	200	0.001	77.16	85.16	86.38	86.6	86.62	86.4	85.6	86	85.74	85.3	84.04	82.96
remove punctuation	1	20000	Adam	100	0.001	74.04	81.22	82.6	84.5	84.78	84.74	84.3	84.56	83.82	84.12	84.848	83.92
remove punctuation	1	5000	Adam	100	0.001	72.3	81.42	83.46	84.4	84.76	84.96	85.04	84.74	84.58	84.36	85.54	84.86

Figure 14: validation accuracy for all parameters tuning

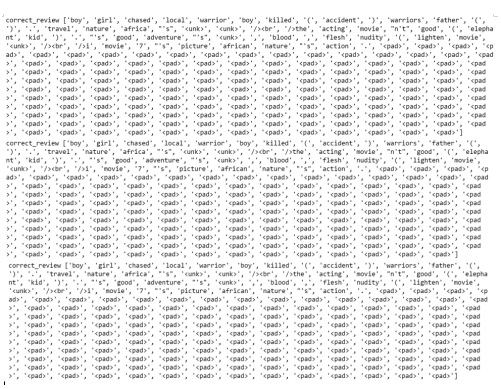

Figure 15: correct reviews

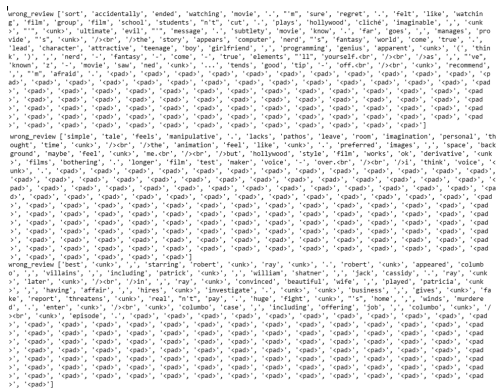

Figure 16: wrong reviews

For the best trained model, I select 3 correct predicted reviews and 3 wrong predicted review. please see table 15 and 16 .

3 Explanation

For all figures, you might see in jupyter notebook and excel in Github. This is my link: urlhttps://github.com/xy990/NLPhw1