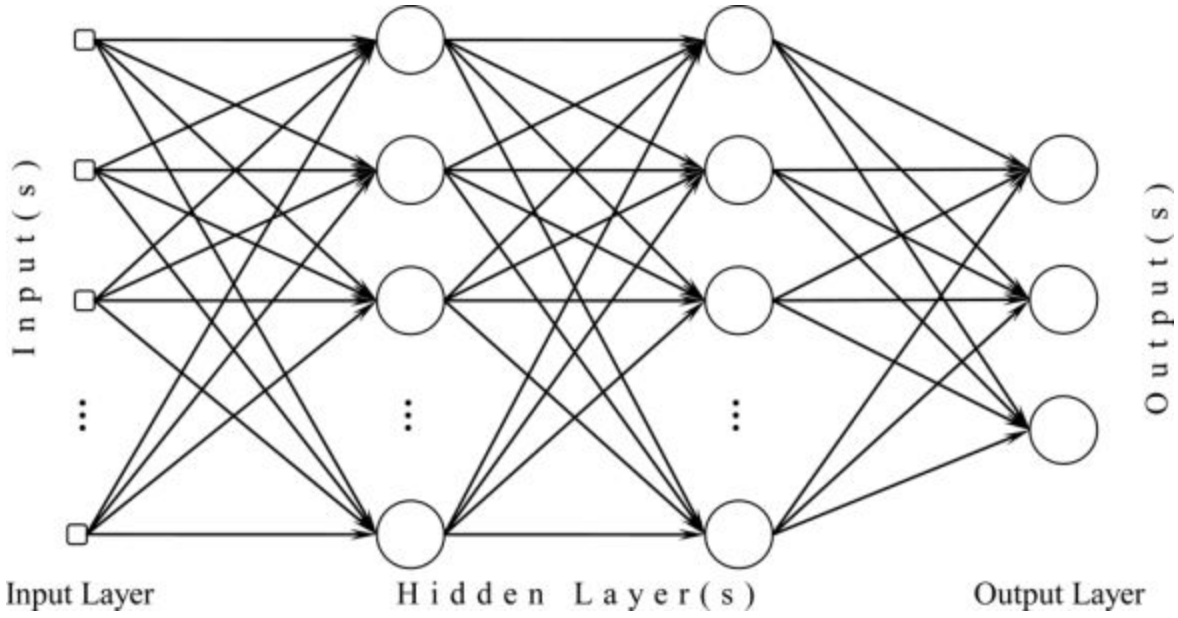
Proper Name and Newsroom Classification

NLP Assignment 1



Xiaohang Lu - xl672

Disheng Zheng - ds336

02.25.2018

Introduction

In this assignment, we build three models of text classifier: perceptron, maximum entropy, multilayer perceptron to classify the proper name and the news group.

On the leaderboard, we have achieved 0.8491, 0.8546, and 0.8382 for propername mlp, maximum entropy, and perceptron model respectively.

We have achieved 0.7681, 0.7570, and 0.7289 for newsgroup mlp, maximum entropy, and perceptron model respectively.

PROCEDURE

Preprocess

Propername

Proper name dataset includes a set of single word or phrase, which belongs to five classes: place, person, drug, company, movie. The task is to take proper names like Eastwood Park and decide whether they are places, people, etc.

To preprocess the data of the proper name, for the label of the class, we assign each class with an id: {'place': 0, 'person': 1, 'drug': 2, 'company': 3, 'movie': 4}. For getting the feature, we apply n-gram model to extract the n-character from each word(phrase). Here we refer the sklearn.feature_extraction.text.CountVectorizer library to fit the training data, and use this fitted vectorizer to transform the training data, development data and testing data to the feature vector. When extracting the n-character from each word, we can set the n a range to extract more feature to find the more predictive n-character.

Newsgroup

The Newsgroup data set, every data is a news article that can be divided into 20 different categories. The article is of a format of an email. There is a header and footer for each which includes an email address, subject, summary, organization, and number of lines. The footer includes email address, and physical address. Then the article often has quotes which feeds great feature for article classification.

I extract the data first by reading the article line by line. I select my feature to be either 1 or 2 grams of words, since words by itself would supply meanings and a phrase of 2

words gives sentiments and contexture information. I limited the dimension of my features to top 20000 most common ones. Then, I eliminated the stop words such as “a”, “the” from the article by using the nltk stopwords library. I then used the stem tokenizer from nltk library to turn words into their basic format such as “developing”, “developed”, “development” -----> “develop”. I then used one hot encoding to turn each unique words into a vector to have an article represented by a sparse matrix.

Perceptron

Perceptron learning algorithm is to predicted a binary label. So for each class, we build its own weight and select the class with highest score as its label which is computed as weight vector * feature vector. Each iteration, if it is the wrong class with the highest score, the weight corresponding to the wrong class should subtract learning_rate * feature vector, and the weight corresponding to the right class should add learning_rate * feature vector. When predicting the label for the test data, we select the class with highest score. Here, we can improve the model through adjusting the learning rate and iteration number.

Propername

We experiment with different learning rate, iteration and n-range of the feature vector.

Different n-range

Learning rate	iteration	n-range	Accuracy on dev
0.01	100	(1,1)	0.4991
0.01	100	(1,2)	0.7369
0.01	100	(1,3)	0.8012
0.01	100	(1,4)	0.8524

Different learning rate and Different iteration

Learning rate	iteration	n-range	Accuracy on dev
0.01	100	(1,2)	0.7369
0.001	100	(1,2)	0.7300
0.01	500	(1,2)	0.7069

0.001	500	(1,2)	0.7245

As we can see from the above figure, adjusting learning rate and iteration has limited improvement on the accuracy of the development data. Even, increasing iteration from 100 to 500 on iteration 100, the accuracy will decrease a little. However, adjusting the n-range has dramatic improvement. But enlarging n-range means enlarging the feature vector length dramatically. So the running time will increase a lot. To balance the running time and accuracy, we set the n-range (1,4) with learning rate 0.01 and iteration 100.

Confusion Matrix

real\predict	place	person	drug	company	movie	sum
place	0.1421	0.0069	0.0066	0	0.0180	0.1736
person	0.0093	0.1535	0.0041	0.0010	0.0173	0.1852
drug	0.0135	0.0021	0.2122	0.0007	0.0093	0.2378
company	0.0003	0.0003	0.0010	0.1113	0.0017	0.1146
movie	0.0252	0.0162	0.0114	0.0024	0.2333	0.2885
sum	0.1904	0.179	0.2353	0.1154	0.2796	1

As we can see from the confusion matrix, for each class, place and movie are the most often misclassified (confused) class, such as Easingwold, Lea Cross, Azzurro, Bajland. Person are often wrongly classified into movie, and drug are often wrongly classified into place.

Newsgroup

Here we experiment with different learning rate, iteration and n-range of the feature vector.

Different n-range

Learning rate	iteration	n-range	Accuracy on dev
0.01	100	(1,1)	0.66
0.01	100	(2,2)	0.58

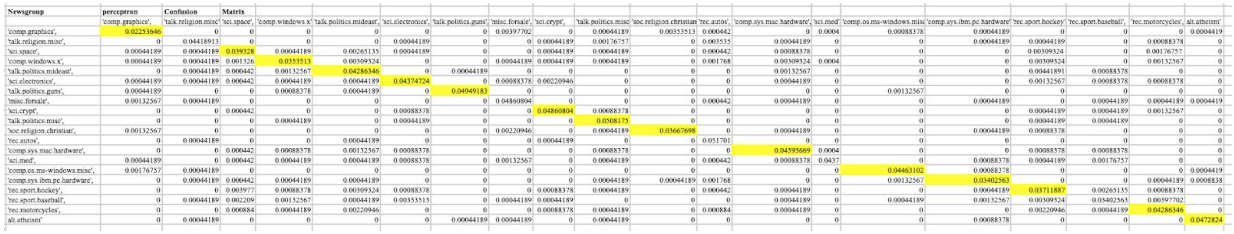
0.01	100	(1,2)	0.86

Different learning rate and Different iteration

Learning rate	iteration	feature limits	Accuracy on dev
0.01	100	10000	0.7475
0.001	100	30000	0.8946
0.01	300	10000	0.7575
0.001	300	30000	0.8860

As we can see from the above figure, adjusting learning rate and iteration has limited improvement on the accuracy of the development data. Even, increasing iteration from 100 to 300 on iteration 100, the accuracy will decrease a little. However, adjusting the n-range has dramatic improvement. But enlarging n-range means enlarging the feature vector length dramatically. So the running time will increase a lot. To balance the running time and accuracy, we set the n-range (1,2) with learning rate 0.01 and iteration 100. Also, notice that when we include more features in the model the accuracy increase drastically.

Confusion matrix



Newsgroup	perceptron	Confusion	Matrix																	
	'comp.graphics',	'talk.religion.misc'	'sci.space'	'comp.windows.x'	'talk.politics.mideast'	'sci.electronics'	'talk.politics.guns'	'misc.forsale'	'sci.crypt',	'talk.politics.misc'	'soc.religion.christian	'rec.autos',	'comp.sys.mac.hardware'	'sci.med'	'comp.os.ms-windows.misc	'comp.sys.ibm.pc.hardware'	'rec.sport.hockey'	'rec.sport.baseball',	'rec.motorcycles',	'alt.atheism'
'comp.graphics',	0.02253646	0	0	0	0	0	0	0.00397702	0	0.00044189	0.00535513	0.000442	0	0.0004	0.00088378	0.00044189	0	0	0	0.0004419
'talk.religion.misc',	0	0.04418913	0	0	0	0.00044189	0	0	0.00044189	0.00176757	0	0.003535	0.00044189	0	0	0.00044189	0.00044189	0	0.00088378	0
'sci.space',	0.00044189	0.00044189	0.039328	0.00044189	0.00265135	0.00044189	0	0	0.00044189	0.00044189	0	0.000442	0.00088378	0	0	0	0.00309324	0	0.00176757	0
'comp.windows.x',	0.00044189	0.00044189	0.001326	0.0353513	0.00309324	0	0	0.00044189	0.00044189	0.00044189	0	0.001768	0.00309324	0.0004	0	0	0.00309324	0	0.00132567	0
'talk.politics.mideast',	0	0.00044189	0.000442	0.00132567	0.04286346	0	0.0004419	0	0	0	0	0	0.00132567	0	0	0	0.00441891	0.00088378	0	0
'sci.electronics',	0.00044189	0.00044189	0.000442	0.00044189	0.00044189	0.04374724	0	0.00088378	0.00220946	0	0	0	0.00044189	0	0	0	0.00132567	0.00088378	0.00088378	0
'talk.politics.guns',	0.00044189	0	0	0.00088378	0.00044189	0	0.04949183	0	0	0	0	0	0	0	0.00132567	0	0	0	0	0
'misc.forsale',	0.00132567	0.00044189	0	0	0	0	0	0.04890804	0	0	0	0.000442	0.00044189	0	0	0.00044189	0	0.00044189	0.00044189	0.0004419
'sci.crypt',	0	0	0.000442	0	0	0.00088378	0	0	0.04860804	0.00088378	0	0	0	0	0	0	0.00044189	0.00044189	0.00132567	0
'talk.politics.misc',	0	0	0	0.00044189	0	0.00044189	0	0	0	0.0508175	0	0	0	0	0	0	0.00044189	0.00044189	0	0
'soc.religion.christian',	0.00132567	0	0	0	0	0	0	0.00220946	0	0.00044189	0.03667698	0	0.00044189	0	0	0.00044189	0.00088378	0	0	0
'rec.autos',	0	0.00044189	0	0	0.00044189	0	0	0	0.00044189	0	0	0.051701	0	0	0	0	0	0	0	0
'comp.sys.mac.hardware',	0	0	0.000442	0.00088378	0.00132567	0.00088378	0	0	0	0.00088378	0	0	0.04595669	0.0004	0	0	0.00088378	0.00088378	0	0
'sci.med',	0.00044189	0	0.000442	0.00044189	0.00044189	0.00088378	0	0.00132567	0	0.00044189	0	0.000442	0.00088378	0.0437	0	0.00088378	0.00044189	0.00176757	0	0
'comp.os.ms-windows.misc',	0.00176757	0.00044189	0	0	0	0	0	0	0	0	0	0	0	0	0.04463102	0.00088378	0	0	0	0.0004419
'comp.sys.ibm.pc.hardware',	0	0.00044189	0.000442	0.00044189	0.00044189	0	0	0	0	0.00044189	0.00044189	0.001768	0	0	0.00132567	0.03402563	0	0	0.00044189	0.0008838
'rec.sport.hockey',	0	0	0.003977	0.00088378	0.00309324	0.00088378	0	0	0.00088378	0.00044189	0	0.000442	0.00044189	0	0	0.00044189	0.03711887	0.00265135	0.00088378	0
'rec.sport.baseball',	0	0.00044189	0.002209	0.00132567	0.00044189	0.00353513	0	0.00044189	0.00044189	0	0	0	0.00044189	0	0.00044189	0.00132567	0.00309324	0.03402563	0.00397702	0
'rec.motorcycles',	0	0	0.000884	0.00044189	0.00220946	0	0	0	0.00088378	0.00044189	0	0.000884	0.00044189	0	0	0	0.00220946	0.00044189	0.04286346	0
'alt.atheism'	0	0.00044189	0	0	0	0	0.00044189	0.00044189	0	0.00044189	0	0	0	0	0	0.00088378	0	0	0	0.047282

Base on the confusion matrix tall.politics.mideast can often be confused with comp.sys.ibm.pc.hardwares. This could be because sometimes, when talking about middle east politics, cybersecurity is often mentioned. The misclassification rate overall is very low.

Maximum Entropy

Maximum entropy is to predict the p(y|x) with the highest probability. Here we realize the training part with opt.fmin_l_bfgs_b function. Within the optimizing function, we mainly realize two calculation. One is to calculate the Σlog p(y|x, w), another is to calculate the gradient of this function, which is the total count of feature in correct

candidates minus the expected count of feature in predicted candidates.

Propername

Here we experiment with different iteration and n-range of the feature vector.

Different n-range

iteration	n-range	Accuracy on dev
100	(1,2)	0.783
100	(1,3)	0.841
100	(1,4)	0.85

Different learning rate and Different iteration

iteration	n-range	Accuracy on dev
50	(1,2)	0.779
100	(1,2)	0.783
150	(1,2)	0.788

As we can see from the above figure, adjusting iteration has limited improvement on the accuracy of the development data. However, adjusting the n-range has dramatic improvement. But enlarging n-range means enlarging the feature vector length dramatically. So the running time will increase a lot. To balance the running time and accuracy, we set the n-range (1,4) with iteration 100.

Confusion Matrix

real\predict	place	person	drug	company	movie	sum
place	0.1358	0.0083	0.0076	0	0.022	0.1737
person	0.0083	0.1455	0.0038	0.0007	0.0270	0.1853
drug	0.0066	0.0024	0.2160	0.0010	0.0118	0.2378
company	0	0.0007	0.0007	0.1100	0.0034	0.1148
movie	0.0218	0.0166	0.0149	0.0010	0.2344	0.2887
sum	0.1725	0.1735	0.243	0.11269	0.2986	1

As we can see from the confusion matrix, for each class, place and movie are the most misclassified class, such as Belgium, Azzurro, Bajland. Person are often wrongly classified into movie.

Newsgroup

Here we experiment with different iteration and feature limits of the feature vector.

iteration	Feature limits	Accuracy on dev
50	10000	0.799
100	10000	0.783
150	30000	0.878

As we can see from the above figure, adjusting iteration has limited improvement on the accuracy of the development data. However, adjusting the feature limits has dramatic improvement. But enlarging features included means enlarging the feature vector length dramatically. So the running time will increase a lot.

Confusion matrix

0.0451	0	0	0	0.0004	0	0.0027	0	0.0009	0	0	0.0009	0	0	0	0	0	0.0018	0.0004	0
0	0.0486	0.0004	0	0.0004	0	0.0004	0	0	0.0004	0	0	0.0004	0	0.0009	0	0	0.0004	0.0004	0
0	0.0004	0.0504	0	0.0004	0	0	0	0	0	0	0	0	0	0	0.0004	0.0004	0	0.0004	0.0004
0	0	0	0.0482	0.0018	0	0	0	0.0004	0	0	0.0004	0.0013	0	0.0004	0	0	0	0.0004	0
0.0022	0	0	0.0004	0.0433	0	0.0017	0	0.0004	0.0004	0	0.0013	0	0	0.0004	0	0	0.0004	0.0004	0.0004
0	0	0	0	0	0.0499	0	0	0.0009	0	0	0.0009	0	0	0	0	0	0	0	0.0009
0.0031	0	0	0.0004	0.0017	0	0.0384	0	0.0027	0	0.0004	0.0013	0	0	0	0	0	0	0.0035	0.0004
0	0	0	0	0	0	0	0.0455	0.0004	0	0.0017	0	0	0	0	0	0.0004	0	0	0
0.0009	0.0013	0	0.0009	0.0022	0.0009	0.0013	0	0.0384	0	0	0.0004	0.0022	0	0	0	0	0.0004	0.0018	0.0013
0	0.0004	0	0	0.0004	0	0.0004	0	0	0.0513	0	0	0	0	0	0	0	0	0	0.0004
0.0004	0	0	0	0	0	0	0.0009	0	0	0.0358	0.0009	0	0.0013	0	0.0004	0	0	0.0004	0.0009
0.0027	0	0	0	0.0009	0	0.0013	0.0004	0	0.0009	0	0.0415	0.0004	0	0.0004	0	0	0.0022	0.0009	0
0.0009	0	0	0.0013	0.0022	0	0.0009	0	0.0018	0	0	0.0004	0.0451	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.0004	0	0.0490	0.0004	0	0	0	0	0
0	0	0	0	0	0	0	0.0004	0.0004	0	0	0.0004	0	0.0004	0.0504	0	0	0.0004	0	0
0	0	0.0022	0.0004	0	0	0.0004	0	0	0	0	0	0	0	0.0004	0.0376	0.0013	0	0	0
0	0	0.0057	0	0	0.0004	0	0.0009	0	0	0.0004	0.0009	0	0	0	0.0022	0.0221	0	0	0.0004
0.0018	0	0	0	0	0	0.0009	0	0.0004	0.0004	0	0.0022	0	0	0.0004	0	0	0.0460	0	0.0004
0.0017	0	0	0	0.0013	0	0.0035	0	0.0009	0	0	0.0009	0	0	0.0004	0	0	0.0013	0.0407	0
0.0004	0.0013	0.0004	0	0.0004	0	0	0	0.0013	0.0004	0.0009	0.0013	0.0013	0	0	0	0	0	0.0004	0.0442

Base on the confusion matrix tall.politics.mideast can often be confused with comp.sys.ibm.pc.hardwares. This could be because sometimes, when talking about middle east politics, cybersecurity is often mentioned. The misclassification rate overall

is very low (<0.03).

Multilayer Perceptron

The multilayer perceptron I designed to have 2 layers with tanh active function. Then a softmax function is applied before the output. The MLP model is built with dynet, the dynamic computational graph framework. Binary log-loss is implemented

$$-\sum(y_i \ln(x_i)+(1-y_i)\ln(1-x_i))$$

Propername

Here we experiment with different iterations and n-range of the feature vector.

Different n-range

iteration	Features Limited	Hidden layer dimension	Accuracy on dev
2	10000	50	0.86
5	20000	64	0.87
10	30000	150	0.83
5	30000	64	0.89

As we can see from the above figure, adjusting iteration has limited improvement on the accuracy of the development data. However, adjusting the limited features has dramatic improvement. But enlarging features means enlarging the feature vector length dramatically. So the running time will increase a lot. To balance the running time and accuracy, we set the n-range (1,4) with iteration 5 and 30000 features with 64 hidden layer dimension.

It's confusion matrix is similar as before. Still, the place and movie are the most misclassified class.

Newsgroup

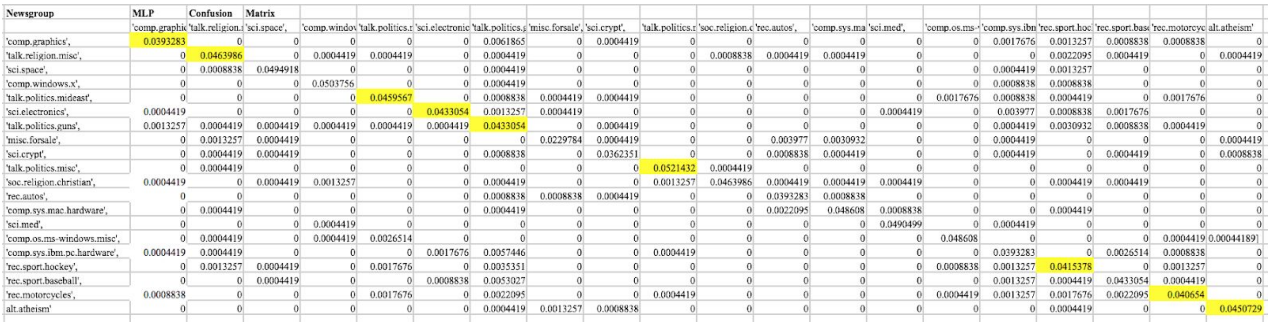
Here we experiment with different iterations and n-range of the feature vector.

Different n-range

iteration	Features Limited	Hidden layer dimension	Accuracy on dev
2	10000	50	0.87
5	20000	64	0.89
10	30000	150	0.85
5	30000	64	0.91

As we can see from the above figure, adjusting iteration has limited improvement on the accuracy of the development data. However, adjusting the limited features has dramatic improvement. But enlarging features means enlarging the feature vector length dramatically. So the running time will increase a lot. To balance the running time and accuracy, we set the n-range (1,4) with iteration 5 and 30000 features with 64 hidden layer dimension.

Confusion Matrix



Newsgroup	MLP	Confusion	Matrix																
	'comp.graphic	'talk.religion.i	'sci.space',	'comp.windov	'talk.politics.r	'sci.electronic	'talk.politics.g	'misc.forsale',	'sci.crypt',	'talk.politics.r	'soc.religion.c	'rec.autos',	'comp.sys.ma	'sci.med',	'comp.os.ms-'	'comp.sys.ibn	'rec.sport.hoc	'rec.sport.bas	'rec.motorcyc	alt.atheism'
'comp.graphics',	0.0393283	0	0	0	0	0	0.0061865	0	0.0004419	0	0	0	0	0	0	0.0017676	0.0013257	0.0008838	0.0008838	0
'talk.religion.misc',	0	0.0463986	0	0.0004419	0.0004419	0	0.0004419	0	0	0	0.0008838	0.0004419	0.0004419	0	0	0	0.0022095	0.0004419	0	0.0004419
'sci.space',	0	0.0008838	0.0494918	0	0	0	0.0004419	0	0	0	0	0	0	0	0	0.0004419	0.0013257	0	0	0
'comp.windows.x',	0	0	0	0.0503756	0	0	0.0004419	0	0	0	0	0	0	0	0	0.0008838	0.0008838	0	0	0
'talk.politics.mideast',	0	0	0	0	0.0459567	0	0.0008838	0.0004419	0.0004419	0	0	0	0	0	0.0017676	0.0008838	0.0004419	0	0.0017676	0
'sci.electronics',	0.0004419	0	0	0	0	0.0433054	0.0013257	0.0004419	0	0	0	0	0	0.0004419	0	0.003977	0.0008838	0.0017676	0	0
'talk.politics.guns',	0.0013257	0.0004419	0.0004419	0.0004419	0.0004419	0.0004419	0.0433054	0	0.0004419	0	0	0	0	0	0	0.0004419	0.0030932	0.0008838	0.0004419	0
'misc.forsale',	0	0.0013257	0.0004419	0	0	0	0	0.0229784	0.0004419	0	0	0.003977	0.0030932	0	0	0.0004419	0	0	0	0.0004419
'sci.crypt',	0	0.0004419	0.0004419	0	0	0	0.0008838	0	0.0362351	0	0	0.0008838	0.0004419	0	0	0.0004419	0	0.0004419	0	0.0008838
'talk.politics.misc',	0	0.0004419	0	0	0	0	0	0	0	0.0521432	0.0004419	0	0	0	0	0	0	0	0	0
'soc.religion.christian',	0.0004419	0	0.0004419	0.0013257	0	0	0.0004419	0	0	0.0013257	0.0463986	0.0004419	0.0004419	0.0004419	0	0	0.0004419	0.0004419	0	0
'rec.autos',	0	0	0	0	0	0	0.0008838	0.0008838	0.0004419	0	0	0.0393283	0.0008838	0	0	0	0	0	0	0
'comp.sys.mac.hardware',	0	0.0004419	0	0	0	0	0.0004419	0	0	0	0	0.0022095	0.048608	0.0008838	0	0	0.0004419	0	0	0
'sci.med',	0	0	0	0.0004419	0	0	0	0	0	0	0	0	0	0.0490499	0	0.0004419	0	0	0	0
'comp.os.ms-windows.misc',	0	0.0004419	0	0.0004419	0.0026514	0	0	0	0	0	0	0	0	0	0.048608	0	0	0	0.0004419	0.00044189
'comp.sys.ibm.pc.hardware',	0.0004419	0.0004419	0	0	0	0.0017676	0.0057446	0	0	0.0004419	0	0	0	0	0	0.0393283	0	0.0026514	0.0008838	0
'rec.sport.hockey',	0	0.0013257	0.0004419	0	0.0017676	0	0.0035351	0	0	0	0	0	0	0	0.0008838	0.0013257	0.0415378	0	0.0013257	0
'rec.sport.baseball',	0	0	0.0004419	0	0	0.0008838	0.0053027	0	0	0	0	0	0	0	0	0.0013257	0.0004419	0.0433054	0.0004419	0
'rec.motorcycles',	0.0008838	0	0	0	0.0017676	0	0.0022095	0	0	0.0004419	0	0	0	0	0.0004419	0.0013257	0.0017676	0.0022095	0.040654	0
alt.atheism'	0	0	0	0	0	0	0.0004419	0.0013257	0.0008838	0	0	0	0	0	0	0	0.0004419	0	0	0.0450729

Based on the confusion matrix, talk.politics.guns is often confused with talk.religion.islam with a rate of 0.004419. This makes sense because many of the terrorist articles contains guns. The misclassification rate overall is very low.

RESULTS

In the end here is the table of models we choose and their respective performances on leader board. The learning rate holds constant lr = 0.01

Propername	ngram	iteration	leaderboard performances

Perceptron	(1, 4)	100	0.8382
Max Entropy	(1, 4)	100	0.8546
MLP	(1, 4)	5	0.8491

Newsgroup	ngram	iteration	leaderboard performances
Perceptron	(1, 2)	200	0.7440
Max Entropy	(1, 2)	100	0.7759
MLP	(1, 2)	5	0.7942

CONCLUSION

We have discovered that for proper names, n-grams ranging from 1 to 4 contributes to classification. For news groups only 1 word or 2 words phrases contributes to classification significantly, multi-gram does not work well, which may because of the sparsity increasing rapidly when increasing n. Also, We must be careful to increase the iteration to avoid overfitting. And, for perceptron and maximum entropy, the accuracy on the development data and test data is similar. But for MLP, the difference of the accuracy is nearly 10%, this is because there are 2000 development set and 7000 test set. Very often, the words contained in test set articles are not included in training set.

The overall misclassification rate in the confusion matrix are lower than 0.004. Often, news articles can not simply be classified into one category. Many of the science articles have overlaps. Moreover, nome news articles about politics also involves science, religion, and gun control. It is reasonable that some of the articles could be misclassified since the labels are very high level.

REFERENCES

1. sklearn.feature_extraction.text.CountVectorizer
2. https://docs.scipy.org/doc/scipy/reference/optimize.html
3. https://pandas.pydata.org/
4. http://www.numpy.org/
5. https://www.nltk.org/

6. https://github.com/clab/dynet