Using pre-trained BERT vectors for text classification

Instructions to run the code

For steps 1 to 4, move submitted scripts and .ipynb file and downloaded bert and data folders in the same folder or location to run everything smoothly.

Steps 1 and 2 – following should be at one place:

1. format_chandni.sh (submitted)
2. run_bert_fv.sh (submitted)
3. bert_vectors_chandni.ipynb (submitted)
4. bert\
   - everything downloaded from git (from repository)
5. uncased_L-12_H-768_A-12\
   - entire folder (downloaded)
6. data\
   1) lang_id_eval.csv (handout)
   2) lang_id_test.csv (handout)
   3) lang_id_train.csv (handout)

Step 3 – following reformatted datafiles are created after running format_chandni.sh

- bert_input_data\
  1) eval.txt
  2) test.txt
  3) train.txt

Step 4 – following feature vectors are created after running run_bert_fv.sh

- bert_output_data\
  1) eval.jsonlines
  2) test.jsonlines
  3) train.jsonlines

For steps 5 to 7, run the bert_vectors_chandni.ipynb file. It will do the following:

Step 5 – train logistic regression and neural network models using train data

Step 6 – make predictions on test data using both models

Step 7 – print evaluation of results from both models

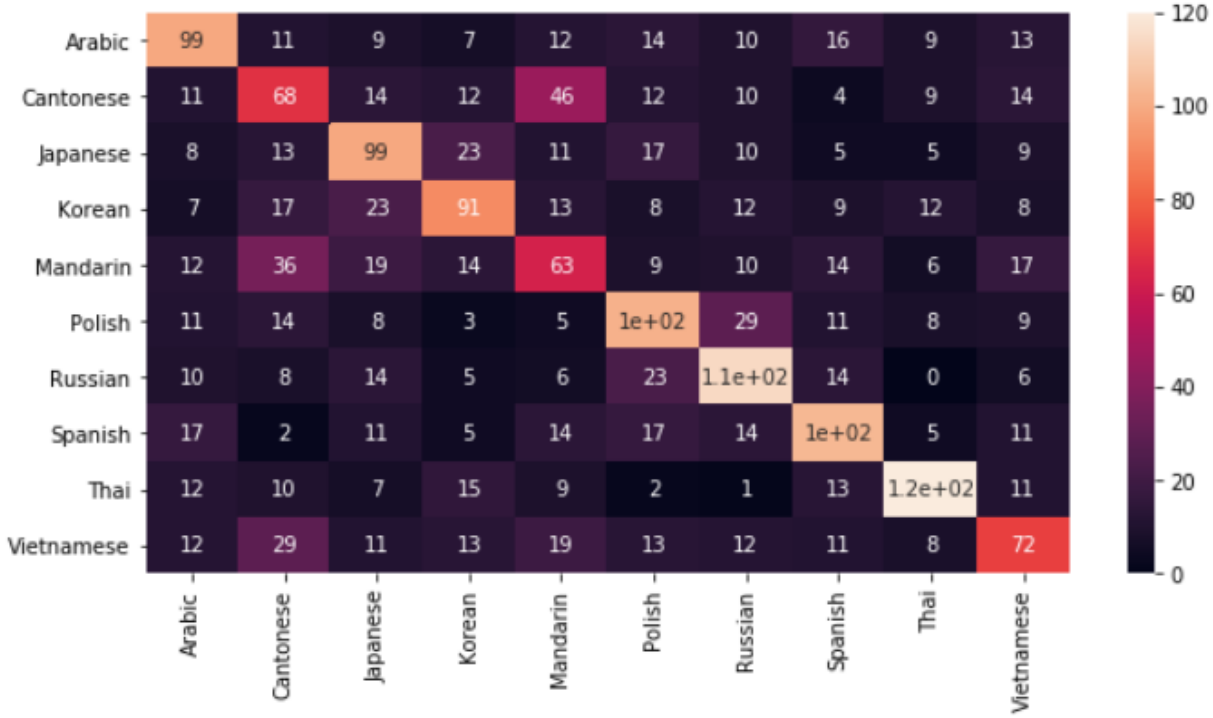
Analysis

Logistic Regression

The accuracy of the model is 47%. The evaluation for each class is as follows:

class	precision	recall	f1-score	support	Misclassification
Arabic	0.50	0.49	0.50	200	10.05
Cantonese	0.33	0.34	0.33	200	13.60
Japanese	0.46	0.49	0.48	200	10.85
Korean	0.48	0.46	0.47	200	10.30
Mandarin	0.32	0.32	0.32	200	13.60
Polish	0.47	0.51	0.49	200	10.65
Russian	0.51	0.57	0.54	200	9.70
Spanish	0.52	0.52	0.52	200	9.65
Thai	0.66	0.60	0.63	200	7.10
Vietnamese	0.42	0.36	0.39	200	11.30

There are total 2000 records in test data and 10 unique classes with 200 records for each. Among all languages, highest precision, recall, and f1-score is for Thai, whereas lowest is for Mandarin. Misclassification is highest for Mandarin and Cantonese, whereas lowest for Thai. Misclassification between each pair of classes is as follows:



	Language	Predicted	Misclassification
0	Russian	Thai	0
0	Thai	Russian	1
0	Spanish	Cantonese	2
0	Thai	Polish	2
0	Polish	Korean	3
0	Cantonese	Spanish	4
0	Japanese	Spanish	5
0	Polish	Mandarin	5
0	Japanese	Thai	5
0	Spanish	Korean	5
0	Spanish	Thai	5
0	Russian	Korean	5
0	Russian	Mandarin	6
0	Mandarin	Thai	6
0	Russian	Vietnamese	6
0	Thai	Japanese	7
0	Korean	Arabic	7
0	Arabic	Korean	7
0	Korean	Polish	8
0	Korean	Vietnamese	8
0	Polish	Thai	8
0	Polish	Japanese	8
0	Russian	Cantonese	8
0	Vietnamese	Thai	8
0	Japanese	Arabic	8
0	Polish	Vietnamese	9
0	Japanese	Vietnamese	9
0	Cantonese	Thai	9
0	Arabic	Japanese	9
0	Thai	Mandarin	9
0	Korean	Spanish	9
0	Arabic	Thai	9
0	Mandarin	Polish	9
0	Cantonese	Russian	10
0	Thai	Cantonese	10
0	Russian	Arabic	10
0	Japanese	Russian	10
0	Arabic	Russian	10
0	Mandarin	Russian	10
0	Spanish	Japanese	11
0	Thai	Vietnamese	11
0	Polish	Spanish	11
0	Spanish	Vietnamese	11
0	Vietnamese	Japanese	11
0	Polish	Arabic	11
0	Arabic	Cantonese	11
0	Vietnamese	Spanish	11
0	Cantonese	Arabic	11
0	Japanese	Mandarin	11
0	Cantonese	Korean	12
0	Cantonese	Polish	12
0	Korean	Thai	12
0	Korean	Russian	12

0	Vietnamese	Arabic	12
0	Arabic	Mandarin	12
0	Vietnamese	Russian	12
0	Mandarin	Arabic	12
0	Thai	Arabic	12
0	Korean	Mandarin	13
0	Japanese	Cantonese	13
0	Thai	Spanish	13
0	Arabic	Vietnamese	13
0	Vietnamese	Korean	13
0	Vietnamese	Polish	13
0	Polish	Cantonese	14
0	Spanish	Russian	14
0	Cantonese	Japanese	14
0	Arabic	Polish	14
0	Spanish	Mandarin	14
0	Cantonese	Vietnamese	14
0	Mandarin	Spanish	14
0	Mandarin	Korean	14
0	Russian	Japanese	14
0	Russian	Spanish	14
0	Thai	Korean	15
0	Arabic	Spanish	16
0	Spanish	Polish	17
0	Mandarin	Vietnamese	17
0	Japanese	Polish	17
0	Spanish	Arabic	17
0	Korean	Cantonese	17
0	Mandarin	Japanese	19
0	Vietnamese	Mandarin	19
0	Russian	Polish	23
0	Japanese	Korean	23
0	Korean	Japanese	23
0	Vietnamese	Cantonese	29
0	Polish	Russian	29
0	Mandarin	Cantonese	36
0	Cantonese	Mandarin	46

According to the results above, Cantonese is misclassified as Mandarin 46 times and Mandarin is misclassified as Cantonese 36 times. Also, Vietnamese is misclassified as Cantonese 29 times and as Mandarin 19 times. Higher misclassification is also seen between these pairs: Polish and Russian, Japanese and Korean.

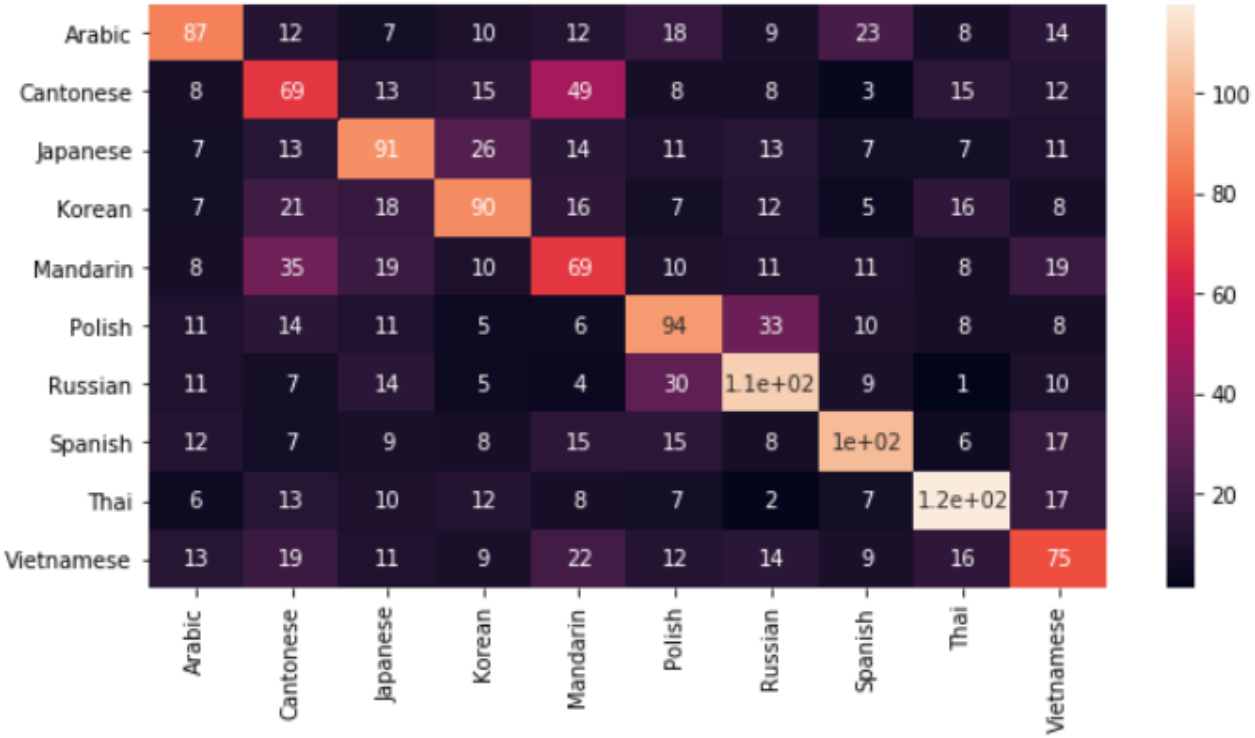
Overall, 1068 records are misclassified and only 932 prediction are correct out of 2000. In search for better accuracy, I'll also train a neural network model. In order to further analyze the problem, I'll be comparing the results of logistic regression model with the results of neural network model.

Neural Network

The accuracy of the model is 45%. The evaluation for each class is as follows:

class	precision	recall	f1-score	support	Misclassification
Arabic	0.51	0.43	0.47	200	9.80
Cantonese	0.33	0.34	0.34	200	13.60
Japanese	0.45	0.46	0.45	200	11.05
Korean	0.47	0.45	0.46	200	10.50
Mandarin	0.32	0.34	0.33	200	13.85
Polish	0.44	0.47	0.46	200	11.20
Russian	0.50	0.55	0.52	200	10.05
Spanish	0.55	0.52	0.53	200	9.05
Thai	0.58	0.59	0.59	200	8.35
Vietnamese	0.39	0.38	0.38	200	12.05

Among all languages, highest precision, recall, and f1-score is for Thai, whereas lowest is for Mandarin. Misclassification is highest for Mandarin, whereas lowest for Thai. Misclassification for all languages, except Arabic and Spanish, has increased when compared to logistic regression model. Therefore, the accuracy is also lower for neural network model. Misclassification between each pair of classes is as follows:



	Language	Predicted	Misclassification
0	Russian	Thai	1
0	Thai	Russian	2
0	Cantonese	Spanish	3
0	Russian	Mandarin	4
0	Russian	Korean	5
0	Korean	Spanish	5
0	Polish	Korean	5
0	Thai	Arabic	6
0	Polish	Mandarin	6
0	Spanish	Thai	6
0	Japanese	Spanish	7
0	Thai	Spanish	7
0	Korean	Polish	7
0	Thai	Polish	7
0	Japanese	Arabic	7
0	Korean	Arabic	7
0	Japanese	Thai	7
0	Russian	Cantonese	7
0	Spanish	Cantonese	7
0	Arabic	Japanese	7
0	Mandarin	Arabic	8
0	Polish	Thai	8
0	Polish	Vietnamese	8
0	Spanish	Korean	8
0	Spanish	Russian	8
0	Thai	Mandarin	8
0	Mandarin	Thai	8
0	Korean	Vietnamese	8
0	Cantonese	Polish	8
0	Arabic	Thai	8
0	Cantonese	Arabic	8
0	Cantonese	Russian	8
0	Vietnamese	Korean	9
0	Russian	Spanish	9
0	Vietnamese	Spanish	9
0	Spanish	Japanese	9
0	Arabic	Russian	9
0	Russian	Vietnamese	10
0	Mandarin	Polish	10
0	Mandarin	Korean	10
0	Polish	Spanish	10
0	Arabic	Korean	10
0	Thai	Japanese	10
0	Mandarin	Russian	11
0	Mandarin	Spanish	11
0	Vietnamese	Japanese	11
0	Russian	Arabic	11
0	Polish	Arabic	11
0	Japanese	Polish	11
0	Polish	Japanese	11
0	Japanese	Vietnamese	11
0	Arabic	Cantonese	12
0	Spanish	Arabic	12

0	Cantonese	Vietnamese	12
0	Korean	Russian	12
0	Arabic	Mandarin	12
0	Thai	Korean	12
0	Vietnamese	Polish	12
0	Thai	Cantonese	13
0	Vietnamese	Arabic	13
0	Cantonese	Japanese	13
0	Japanese	Cantonese	13
0	Japanese	Russian	13
0	Arabic	Vietnamese	14
0	Polish	Cantonese	14
0	Vietnamese	Russian	14
0	Japanese	Mandarin	14
0	Russian	Japanese	14
0	Cantonese	Thai	15
0	Spanish	Mandarin	15
0	Spanish	Polish	15
0	Cantonese	Korean	15
0	Vietnamese	Thai	16
0	Korean	Thai	16
0	Korean	Mandarin	16
0	Thai	Vietnamese	17
0	Spanish	Vietnamese	17
0	Arabic	Polish	18
0	Korean	Japanese	18
0	Mandarin	Vietnamese	19
0	Mandarin	Japanese	19
0	Vietnamese	Cantonese	19
0	Korean	Cantonese	21
0	Vietnamese	Mandarin	22
0	Arabic	Spanish	23
0	Japanese	Korean	26
0	Russian	Polish	30
0	Polish	Russian	33
0	Mandarin	Cantonese	35
0	Cantonese	Mandarin	49

According to the results above, Cantonese is misclassified as Mandarin 49 times and Mandarin is misclassified as Cantonese 35 times. Polish is misclassified as Russian 33 times and Russian is misclassified as Polish 30 times. Arabic is misclassified as Spanish, Japanese as Korean, whereas Korean as Cantonese over 20 times. Also, Vietnamese is misclassified as Cantonese and as Mandarin.

Overall, 1095 records are misclassified and only 905 prediction are correct out of 2000. Most of the misclassification patterns are same in both models, which shows that certain pairs of languages have higher chances of being misclassified as each other.

Improvements

The logistic regression model can be improved by hyperparameter tuning by grid search. The neutral network model can be improved by using hyperparameter optimization tools on parameters like hidden_layer_sizes, activation, solver, alpha, learning_rate, max_iter, etc. Use BERT vectors and more data to train the models in order to see improvements.