Machine Learning for NLP

Part 1

The dataset contains a number of languages with 1000 sentences per language, which means all classes are balanced. These languages range across a number of families including not latin languages such as Arabic, Japanese etc and prove to be a challenge.

Furthermore as the initial train test split is 50:50 we modify this to include greater training data i.e 80:20 training and testing respectively. We furthermore incorporate some linguistic features into our training dataset such as average word length and sentence length to extend the feature space.

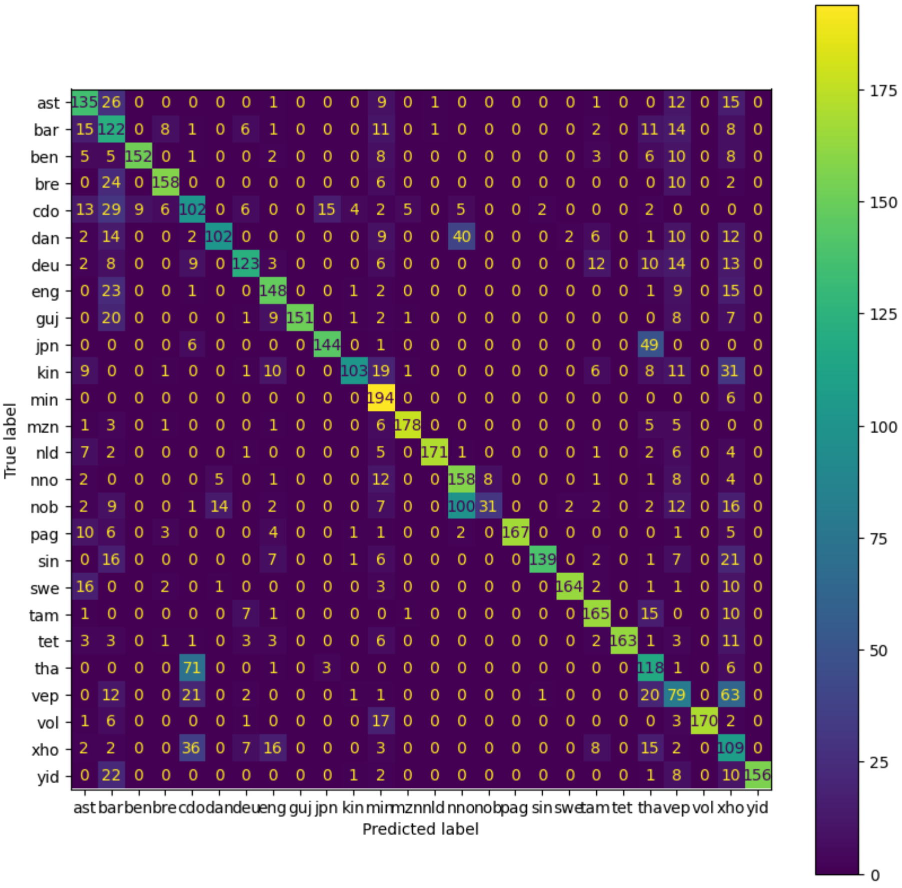
Once we have the dataset ready we use count vectoriser and tf-idf to create a feature vector from the sentences. We then use logistic regression to predict the language. Using grid search we then search for the optimal hyper-parameters. We experiment with the following:

- L2 and none
- LBFGS solver (due to computational constraints we only chose 1)
- Different n-gram vectors (1,1) and (1,2)

Using the best pipeline we then look at the best features for logistic regression for two languages: . These hyper parameters are:

- No penalty
- LBFGS solver
- Ngram range of (1,1)

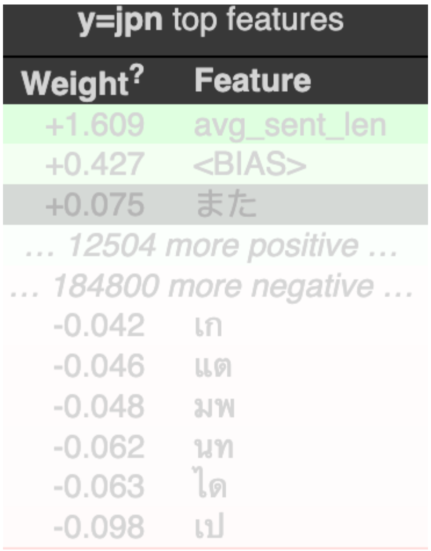
Based on the confusion matrix we get from the best model we see our model achieve an accuracy of nearly 70%. Furthermore we see that we have very high precision and recall as well.



Confusion Matrix

Furthermore we also take a look

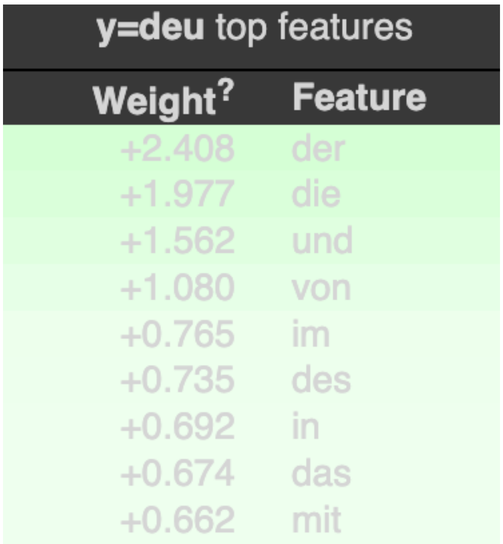
at the logistic regression weights to understand the importance of features. We notice that certain words are very important to detect a language and the linguistic features are not considered that important. Specifically the vectoriser output is considered very important while the extended features are not considered that important



y=jpn top features

Weight?	Feature
+1.609	avg_sent_len
+0.427	<BIAS>
+0.075	また
… 12504 more positive …	
… 184800 more negative …	
-0.042	เก
-0.046	แต
-0.048	มพ
-0.062	นท
-0.063	ได
-0.098	เป

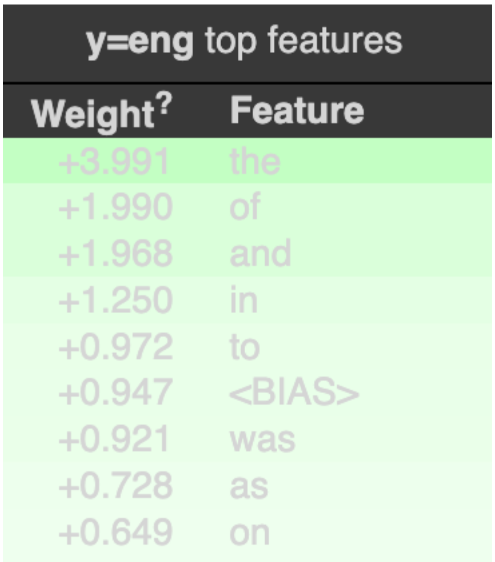
Top Japanese Features



y=deu top features

Weight?	Feature
+2.408	der
+1.977	die
+1.562	und
+1.080	von
+0.765	im
+0.735	des
+0.692	in
+0.674	das
+0.662	mit

Top German Features



y=eng top features

Weight?	Feature
+3.991	the
+1.990	of
+1.968	and
+1.250	in
+0.972	to
+0.947	<BIAS>
+0.921	was
+0.728	as
+0.649	on

Top English features

Our ablation study showed that focusing on the first 500 characters is enough but that reducing the sentences to the first 100 chars or less leads to problems.

Part 2

The dataset and its processing remains as before but without any additional linguistic features. We experiment with different architectures i.e number of layers and nodes per layer, activation functions, vectorisers and early stoping. Due to computational and time constraints we do not perform grid search but instead opt for a greedy approach i.e choose the best option from a single hyper-parameter and then modify the next one.

- We find that ReLu works better than Tanh and Sigmoid activation functions.
- Furthermore we keep our network with three layers of 500 x 250, 250 x 100 and finally 100 x 26. Increasing the number of layers increased the time but did not improve the performance significantly, while reduced layers decreased the accuracy.
- We try a range of learning rates and decide on 0.1 which yields the lowest validation accuracy.
- We also find that early stopping is better for this approach as training for longer yields to overfitting.

Using neural networks and an input feature vector of size 500 we get an accuracy of 97.2 %.