Jade Oakes
COSI 114a
12/14/23

Language Identification

Introduction

I chose to do language identification because I am interested in languages and tasks within the multilingual space. I have worked as a tutor for foreign languages, and I have experimented with various apps for language learning. The task of language identification is relevant for the creation and improvement of language learning apps and technologies.

Data

The dataset I am using is from Kaggle. It was extracted from the Wikipedia language identification benchmark dataset (WiLI-2018). Each row contains a paragraph and its respective language. The columns are “Text” and “language”. The original dataset contains 235 languages, however the dataset from Kaggle has been reduced to 22 languages. I further reduced the dataset to include 11 languages: Dutch, English, Estonian, French, Indonesian, Latin, Portuguese, Romanian, Spanish, Swedish, Turkish. I selected these languages because they use the Latin script. Each language has 1000 rows of data, giving my selected dataset 11,000 rows in total. I split the dataset into train/dev/test. The split is 80/10/10. Below is a sample of the dataset:

Text	**language**
association de recherche et de sauvegarde de lhistoire de roissy-en-france arshrf roissy-en-france ...	French
walter kaudern dalam bukunya menyatakan wawo lage dianggap sebagai desa asal mereka desa ini sendiri...	Indonesian
en navidad de poco después de que interpretó la canción en francés película papillon toi qui regard...	Spanish
sebes joseph pereira thomas på eng the jesuits and the sino-russian treaty of nerchinsk the diary ...	Swedish

Dev Set Results

The models I ran include Multinomial Naïve Bayes, Logistic Regression, and Random Forest. I experimented with three feature sets: unigrams, characters, and character bigrams. The features are binary. I ran many tests on the dev set, experimenting with how changing different hyperparameters affected the accuracy. I ran the tests on each feature set to compare them. Below are some of the results:

Dev set results						
	Unigrams		**Characters**		**Character bigrams**	
Model	α	Accuracy	α	Accuracy	α	Accuracy
Naïve Bayes	0.1	98.80	0.1	88.40	0.1	97.50
Naïve Bayes	0.5	98.50	0.9	88.60	0.8	97.70
Naïve Bayes	501	96.80	501	73.90	501	94.90
	C		C		C	
Logistic Regression	1.0	**99.10**	0.1	89.80	0.1	98.50
Logistic Regression	5.0	**99.10**	0.5	91.00	0.5	**98.60**
Logistic Regression	9.0	99.00	1.0	90.80	1.0	98.60
	n_estimators		n_estimators		n_estimators	
Random Forest	10	98.50	10	89.90	10	96.50
Random Forest	60	98.40	60	91.30	60	98.40
Random Forest	200	98.60	200	91.20	200	98.30
	n_estimators = 200		n_estimators = 60		n_estimators = 300	
	max_depth		max_depth		max_depth	
Random Forest	5	98.10	5	89.20	5	97.70
Random Forest	30	98.50	20	**91.70**	25	98.50
Random Forest	50	98.40	50	91.20	50	98.40

Test set results

A summary of the results of running each model on the test set is below:

Test set results						
	Unigrams		**Characters**		**Character bigrams**	
Model	α	Accuracy	α	Accuracy	α	Accuracy
Naïve Bayes	0.1	97.50	0.9	86.50	0.8	96.40
	C		C		C	
Logistic Regression	1.0	**97.80**	0.5	89.10	0.6	97.60
	n_estimators,		n_estimators,		n_estimators,	
	max_depth		max_depth		max_depth	
Random Forest	200, 30	97.00	60, 20	89.20	60, 30	97.20

The full results from testing the Logistic Regression model on the test set are below:

Logistic Regression			
Unigrams	**C = 1.0**		
	Precision	Recall	F1
Dutch	100.00	96.00	98.00
English	92.00	99.00	95.00
Estonian	96.00	97.00	96.00
French	95.00	100.00	97.00
Indonesian	99.00	97.00	98.00
Latin	98.00	95.00	96.00
Romanian	100.00	99.00	99.00
Spanish	100.00	97.00	98.00
Swedish	100.00	100.00	100.00
Turkish	100.00	98.00	99.00
Accuracy			97.80
Macro Avg	98.00	98.00	98.00
Weighted Avg	98.00	98.00	98.00

Discussion

The models performed similarly on each of the feature sets I tested them on. This could be due to the amount of data in each “Text” column in the dataset. With 1000 rows of data per language, the models were able to consistently get high scores for accuracy. The unigram and character bigram feature sets performed the best. This makes sense because it is less common for a unigram to exist in text from multiple languages. This is also true of character bigrams. There is a greater number of unigrams and character bigrams than there are characters. The character feature set did not perform as well because many languages share the same set of characters. There are some languages that use characters not used by any other language in this dataset. By reducing the dataset to languages with Latin scripts, this reduced the number of languages with unique characters. When experimenting with unigrams, certain languages, such as Swedish and Romanian, had scores of almost 100 for precision, recall, and f1. Overall, the model worked very well, especially with unigram and character bigram feature sets.

Conclusion

All three of the models performed very well on the unigram and character bigram feature sets. Logistic regression with a C value of 1.0 performed the best, with an accuracy of 97.80. Dutch, Romanian, Spanish, Swedish, and Turkish each had a precision score of 100 when given a unigram feature set. It was difficult to decide which features to use, and experimenting on all three took a significant amount of time. It was also difficult to know which hyperparameters to alter, and to interpret the effects that changing them had on the accuracy. Since there was so much data for the models to train on, I would consider reducing the amount of data. This would help the models run faster, and I might be able to better tune the hyperparameters. I would also consider using only the character bigrams feature set because it seems like it gives the most reliable results.