Spanish Dialect Classification

https://github.com/eduardoberford/Spanish-Dialects-Detection

Marco Liconti mliconti3@gatech.edu
Eduardo Amorim eamorim3@gatech.edu
Matt Steele msteele37@gatech.edu
Austin Rubinger arubinger3@gatech.edu

Abstract

This paper introduces an innovative method for classifying distinct Spanish dialects spoken in Spain. Our methodology centers around a comparative analysis involving three distinct models: Random Forest, Naive Bayes Classifiers, and Logistic Regression. Each model underwent training using scaled word-level TF-IDF feature vectors. Then we compared these models to a SOTA model, OpenAI's GPT-4. Through experimentation, we found that Logistic Regression emerged as the most effective for this task, outperforming Random Forest, Naive Bayes Classifiers, and ChatGPT.

1 Introduction

Language Identification represents an important task in Natural Language Processing as there are many real-world applications such as machine translation and chat-bots. There have been many similar projects conducted with language classification between linguistically distant languages, however, classifying dialects seemed to be a more interesting and challenging task due to the shared features and mutual intelligibility between dialects of the same language. Specifically, we seek to classify five Spanish dialects: Asturian, Basque, Galician, Aragonese, and Judeo-Spanish. While Basque is not a dialect of Spanish, it is still spoken in Spain and we decided to include it in our data as a baseline. Our data set consists entirely of Wikipedia pages in each of the five dialects which we will train on a Logistic Regression, Naive Bayes Classifier, and Random Forest models, then analyze their performance and errors using F-1 scores and a confusion matrix.

2 Previous work

There are many previous works that this exploration builds on top of. The first is a paper about classifying the Uralic languages from each other as well as other non-related languages, using the HeLI method which is similar to a Naive Bayes Classifier.[1] The authors used F-1 scores and a confusion matrix to analyze their results. Another paper classified four Indonesian languages using deep learning techniques such as CNN's but was analyzing speech as data rather than text.[2]

3 Dataset

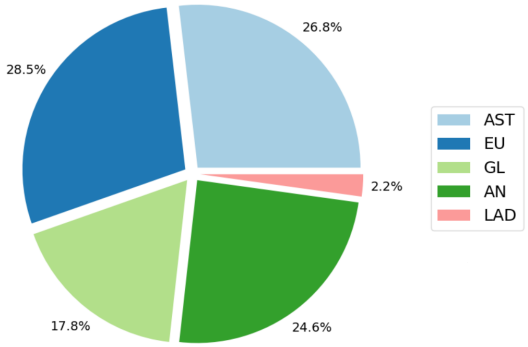

Figure 1: Proportion of Articles by Language

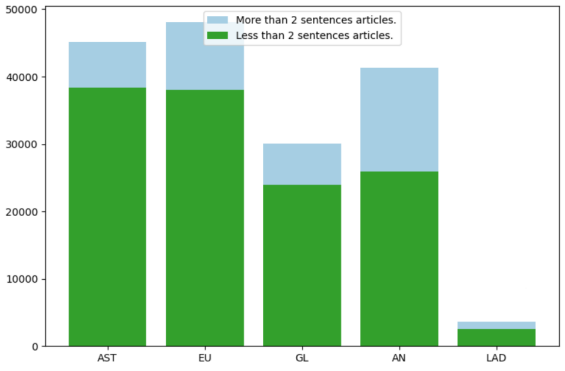

Figure 2: Number of Articles for each Language

[2] Fathoni, R., Salamah, S., et al. Spoken language identification on 4 Indonesian local languages using deep learning. ResearchGate.

[2] Jauhiainen, T., et al. (2020). Uralic Language Identification (ULI) 2020 shared task dataset and the Wanca 2017 ACLAnthology

In this investigation, we utilize a unique dataset comprising of data extracted from Wikipedia pages. The focus of this data is using the same pages in various dialects of Spanish in order to have adjacent lingual dialect comparisons.

The data for this investigation was gathered as follows. We used Wikimedia dumps to be able to get all of the text data of each respective wikipedia. From there, we created a python script that took the downloaded data and turned it into a proper CSV of labeled data. This was done by splitting the data into sentences, and labeling each sentence with a certain label value of whatever language it was. The fact that each dialect has its entire wikipedia as the text data ensures that the data spans a diverse breadth of vocabulary, giving it a comprehensive representation on each dialect.

In this investigation, we initially had 7 dialects, Asturian, Basque, Galician, Aragonese, Judeo-Spanish, Catalan and Spanish. Due to the disproportionate size of the Spanish and Catalan Wikipedias, we chose to exclude these from our dataset intentionally to avoid class imbalance.

4 Empirical Approach

4.1 Model selection

Multiple models were used for the eventual evaluation of the dataset. These included the following models. We used Logistic Regression because of how it performed as a classifier in another paper classifying Italian dialects.[3]Similarly, we used Naive Bayes because of its performance in the aforementioned paper that classified Uralic languages. Additionally, Logistic Regression can be prone to over-fitting when dealing with large data sets, while Naive Bayes is not. Lastly, we used Random Forest because they are relatively easy to implement and good at handling sparse data sets such as the one that we were dealing with.

4.2 Data Processing for Model training

The Data processing for model training was relatively straightforward. After processing the data into a labeled CSV as described in the dataset section, we used 30% of our dataset for evaluation and the other 70% for training. The dataset was processed into vectors for the Random Forest and Naive Bayes classifier models using TF-IDF, and

[3]Camposampiero, G., et al. The Curious Case of Logistic Regression for Italian Languages and Dialects Identification ACLAnthology

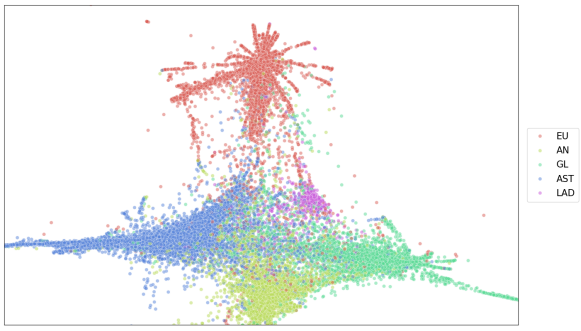

Figure 3: TFIDF Vector Representation of Data

the Logistic Regression model did this built out of the box, so we passed it the labeled text data. Below is a visual breakdown of the vectorized text data from our investigation. You can see each of the dialects clustering together.

	precision	recall	f1-score	support
0	0.98	0.99	0.98	10872
1	0.99	0.99	0.99	12376
2	0.98	0.98	0.98	8008
3	0.98	0.97	0.97	5562
4	0.99	0.90	0.95	694
accuracy			0.98	37512
macro avg	0.99	0.97	0.98	37512
weighted avg	0.98	0.98	0.98	37512

Figure 4: Logistic Regression F1 Scores

4.3 Logistic Regression Results

The logistic regression model performed very well with the data sets. The macro average for precision was 0.99 and the macro average for recall was 0.97 indicating that, on average, of all the articles predicted to be their respective languages, 99% of them were accurately predicted, and that of all the articles that were the class in question, 97% were correctly predicted. For the Logistic Regression model, Judeo-Spanish has the lowest F-1 Score of .95, which is still very high. The diagonals of the confusion matrix show us that all of the languages performed very well, but Judeo-Spanish performed the lowest with only 90% of the cases being predicted correctly. Ladino was most commonly mistaken with Asturian.

4.4 Naïve Bayes Net Results

The Naive Bayes Net's model macro average for precision was 0.94 and the macro average for recall was 0.94. Aragonese has the lowest F-1 Score of .93. The confusion matrix shows that all of the

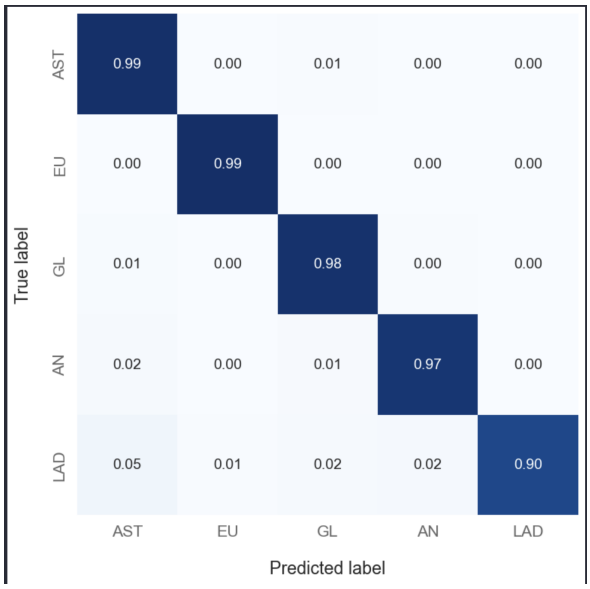

Figure 5: Logistic Regression Confusion Matrix

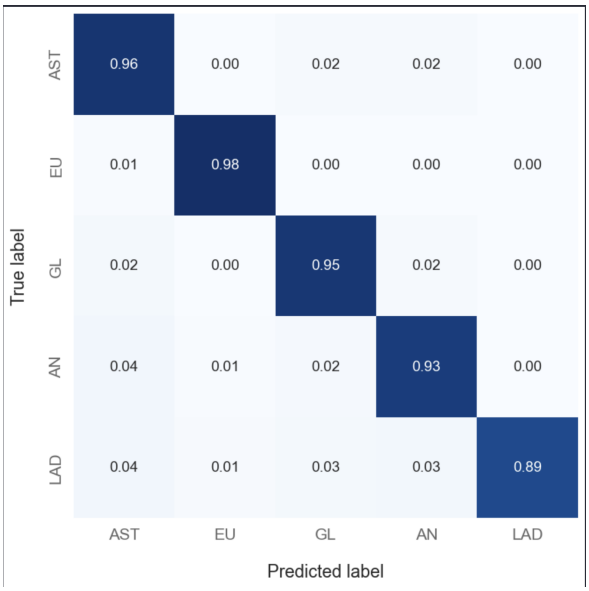

Figure 7: Naïve Bayes Confusion Matrix

	precision	recall	f1-score	support
0	0.95	0.96	0.96	10872
1	0.99	0.98	0.99	12376
2	0.95	0.95	0.95	8008
3	0.92	0.93	0.93	5562
4	0.89	0.89	0.89	694
accuracy			0.96	37512
macro avg	0.94	0.94	0.94	37512
weighted avg	0.96	0.96	0.96	37512

Figure 6: Naïve Bayes Net F1 Scores

	precision	recall	f1-score	support
0	0.92	0.96	0.94	10872
1	0.92	1.00	0.96	12376
2	0.98	0.92	0.95	8008
3	0.99	0.85	0.91	5562
4	1.00	0.61	0.76	694
accuracy			0.94	37512
macro avg	0.96	0.87	0.90	37512
weighted avg	0.94	0.94	0.94	37512

Figure 8: Random Forest F1 Scores

languages performed well, but Judeo-Spanish performed the lowest with only 89% of the cases being predicted correctly. Judeo-Spanish and Basque were most commonly mistaken with Asturian. This model then ranks lower than Logistic Regression.

4.5 Random Forest Results

Random Forest performed the worst of our experiments with a 98% macro avg F-1 score. The main problem with it was Judeo-Spanish again, with only .61 accuracy. The confusion matrix tells us that Judeo-Spanish was incorrectly labaled as Asturian. This might sugges that random forest and decision tree models are more susceptible to class imbalance due to the fewer number of inputs for Judeo-Spanish.

4.6 GPT 4 Prompting Results

As a baseline comparison for the other models, we chose to use GPT-4-Turbo and prompt it to try and classify these texts. The prompt was developed first through some prompt engineering and testing to confirm its efficacy, and was then run on a sampled 1000 data points from our training set. This was due to the fact that the API calls are simply expensive, and due to technical difficulties our Fine Tuned LLama 2 instance did not work, so we used this instead.

The results from GPT-4 were not nearly as good as the other models we did, despite the fact it is state of the art. As shown, it only had a 0.74 overall recall score. This is lower than all of our prior models. In fairness to it, this was not a Fine-Tuned model for classification, and the results reflect this.

5 Overall Findings

Overall we found that the order of performance of the four models we trained was: 1. Logistic Regression, 2. Naive Bayes, 3. Random Forest, 4. GPT4. For the first three models, the three languages with the most data associated with them, Asturian, Basque, and Galician all maintain high levels of accuracy so the differences in performances

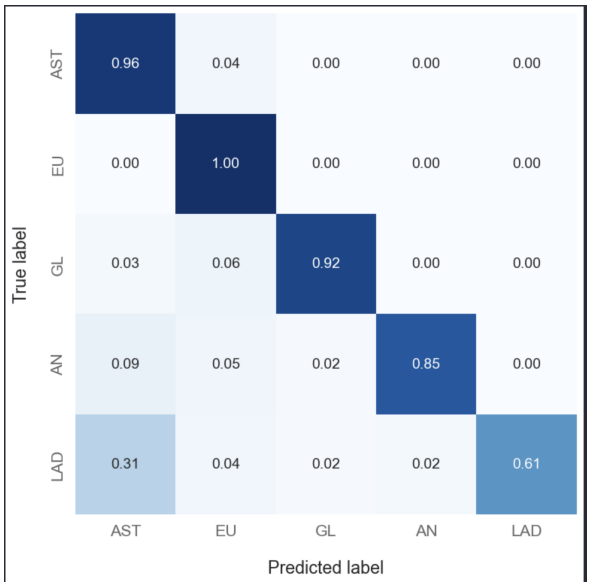

Figure 9: Random Forest Confusion Matrix

	precision	recall	f1-score	support
0	0.53	0.90	0.67	200
1	0.68	0.29	0.41	200
2	0.94	0.83	0.88	200
3	0.92	0.76	0.83	200
4	0.87	1.00	0.93	200
accuracy			0.76	1000
macro avg	0.79	0.76	0.74	1000
weighted avg	0.79	0.76	0.74	1000

Figure 10: GPT-4 Prompting F1 Scores

come down to Ladino and Aragonese, indicating that Logistic Regression was able to handle the disparities of language proportions better than the other models. Interestingly, the predicted state-of-the-art model, GPT4, performed worse than every other model. Our GPT4 instance was only being prompted, and was tested on an equal sample from each language and performed the best on Ladino, the smallest sample, and the worst on the second most abundant and most linguistically distant sample Basque. There is a possibility that the model was overfitted in its pre training for Ladino because the subset of articles written in Ladino on Wikipedia is small and covers specific topics. The GPT-4 Prompting did not result in a great outcome, however it is likely with some fine tuning that this would be the best option. Due to the cost, fine tuning an OpenAI pre trained transformer fell outside the scope of our project.

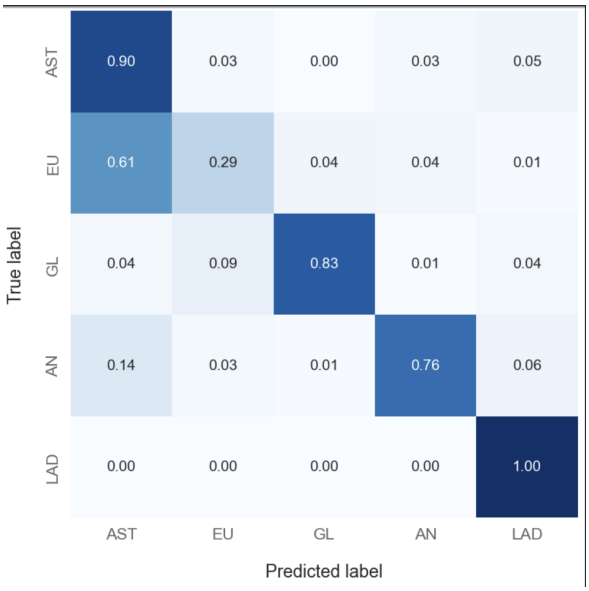

Figure 11: GPT-4 Prompting Confusion Matrix

Limitations

The results of this paper seem extraordinarily accurate given the task, but we feel that this may be misleading. In the context of a language classification task, this high accuracy may likely be due to over-fitting. Wikipedia is meant to be an informational site, so the structure of Wikipedia articles is quite formulaic, and the grammar is mostly going to be standard across a dialect's site. For these reasons, the models may have overfitted to the nuances of the Wikipedia data, which made them good at predicting the Wikipedia data they weren't trained on, but may impact their performance on this task in a wider context.

This extra, less structured, dialog context is especially prevalent in an online and offline world in which the structural and grammatical variation in a singular dialect is much greater, and likely would include slang terms. Future investigation is required to judge the efficacy of such a simple model like logistic regression on a full world of dialect data.

One other important note to mention is that the vectorizing of the data likely has a significant influence on the problem as a whole. This is because vectors can give interesting relational qualities to text data given the embedding method. Using TFIDF Vectorization does not explicitly group by dialect with language, we can see in our dataset vector visualization that the same dialects cluster together very closely.

Lastly, our data set was not equally sized as there

was about twenty times as much support for the most abundant data set, Basque as there was for the least abundant data set, Judeo-Spanish. Having an unequal distribution of data could lead to a bias towards the majority class.

Team Contributions

The team split the project into many pieces over the course of its completion in order to most equitably and effectively distribute the work.

Marco specialized in the domain research and intitial dataset setup of the data. Marco as the deciding factor for the actual topic choice as he collaborated closely with Austin to find the most interesting problem to tackle. He also contributed heavily to the report, and the choice of models to be used.

Eduardo wrote the code that visualized our dataset based on all of its interesting quirks and features. This allowed for a comprehensive overview of what the dataset looked like and allowed us to have some level of baseline understanding of what the results would be. He also wrote the initial data processing code for the models, and wrote the logistic regression code.

Austin worked on analyzing the results of the models based on the code written by Eduardo and Matt. Austin was especially critical in the background domain research of the problem space, and his analysis was extremely thorough on the model outputs. Austin also discretely analyzed many limitations of our work, and pointed out future directions that we could take it.

Matt contributed by parsing the initial data from the wikipedia pages by usng the wikimedia dumps, and then taking those and converting them into a large CSV that enabled the training efforts. He also wrote code for the models and some of their analysis, with the main contributions being the Random Forest and Naive Bayes models in code, along with some of the code that supported the analysis.

References

Fathoni, R., Salamah, S., et al. Spoken language identification on 4 Indonesian local languages using deep learning. ResearchGate.

Jauhiainen, T., et al. (2020). Uralic Language Identification (ULI) 2020 shared task dataset and the Wanca 2017 ACLAnthology

Camposampiero, G., et al. The Curious Case of Logistic Regression for Italian Languages and Dialects Identification ACLAnthology