Autumn Sehy

12/14/2023

Classifying Folktales by Origin using Ecological Features

Stories describe the world around us: the cross-cultural myth of the coyote teaches rose hip toxicity across the American Plains, the Golem is used in modern Jewish lore to approach the epigenetics of trauma, and the festival of Tanabata marks the end of the monsoon season through the annual crossing of Vega and Altair in the sky.

This method of information-sharing is difficult to extract data from. While some tales explain how to stun a bear through pulling on its uvula, others declare that bears live in cottages and drink tea. I wanted to use my anthropology, literature, and naturalist backgrounds to explore folklore data with a quantitative approach to try to ease this issue.

To do this I began with the Kaagle folktales database. This database is mostly data from fairytalez.com, a site that calls themselves the "world's largest fairy tale database". It has 2838 folktales classified across 57 nations.

Due to living in Japan I have a bias towards Japanese Yokai stories, and a curiosity about stories crossing land-water borders from India and China. Therefore I chose labels with the most instances in the Asia/Pacific region. They ended up being: Japanese, Chinese, Indian, Philippine, Arabic, Russian, and Australian_Ethnic.

I stuck to bigrams, trigrams, and mammal features to begin. I split the data into 80/10/10 spread: in total there are 449 stories in train, 56 stories in test, and 57 stories in dev. The proportion in train, which was mimicked in test and dev was: 102 Indian, 82 Chinese, 69 Japanese, 63 Russian, 57 Filipino, 45 Australian ethnic, and 31 Arabic stories. Having so few stories for test and dev did produce problems when classifying only on mammal features, so I decided to add in a trees plus mammals feature.

To extract the mammal features I used the Mammal Diversity Database, which is a database ran by the American Society of Mammologists in conjunction with Arizona State University. This database has the scientific and common names in it, but the common names are not the same as "layman common names". Therefore I made a new column in the database and extracted simple names from each animal. Most were easy enough, ex: for ring-tailed lemur I wrote lemur. With 6650 species of mammals this seemed to be a monumental task at first, but about 1/3 of the database ended up being different species of mice, which greatly reduced the workload.

The trees are from a database of simplified names of trees I've made. Classifying trees is a difficult task, and I tackled this issue by mixing anything an ecologist or layperson would classify as a tree, which means I mostly mashed shrubs, trees, and palms in one category.

To classify I used Scikit learn multinomial naive bayes, complement bayes, logistic regression, and random forest classifiers. For each classifier I tested three different configurations, with a total of 45 tests.

Dev Set Results

Classifier	Tuning	Accuracy *one reported number if accuracy was the same across tests
Multinomial Naive Bayes Bigrams	alpha = 1.0, 0.5, 0.05	56, 68, 79
Multinomial Naive Bayes Trigrams	alpha = 1.0, 0.5, 0.05	60
Multinomial Naive Bayes Mammals	alpha = 1.0, 0.5, 0.05	19
Multinomial Naive Bayes Mammals and Trees	alpha = 1.0, 0.5, 0.05	37
Complement Naive Bayes Bigrams	alpha = 1.0, 0.5, 0.05	82
Complement Naive Bayes Trigrams	alpha = 1.0, 0.5, 0.05	82
Complement Naive Bayes Mammals	alpha = 1.0, 0.5, 0.05	9
Complement Naive Bayes Mammals and Trees	alpha = 1.0, 0.5, 0.05	26
Random Forest Bigrams	Max_depth = standard, 2, 4	56
Random Forest Trigrams	Max_depth = standard, 2, 4	51
Random Forest Mammals	Max_depth = standard, 2, 4	19
Random Forest Mammals and Trees	Max_depth = standard, 2, 4	37

Logistic Regression Bigrams	C= 1, 10, 100
	Class weight = standard, balanced, balanced
	Iters= standard, 1000, 1000
		84

Logistic Regression Trigrams	C= 1, 10, 100
	Class weight = standard, balanced, balanced
	Iters= standard, 1000, 1000
		72

Logistic Regression Mammals	C= 1, 10, 100
	Class weight = standard, balanced, balanced
	Iters= standard, 1000, 1000
		19

Logistic Regression Mammals and Trees	C= 1, 10, 100
	Class weight = standard, balanced, balanced
	Iters= standard, 1000, 1000
		33

For each test I tried configuring hyperparameters, but the only place they made a difference was for Multinomial Naive Bayes.

For the final test set we can look at a similar table of configurations. Hyperparameter tuning didn't change any results here.

Test Set Results

Classifier	Tuning	Accuracy *one reported number if accuracy was the same across tests
Multinomial Naive Bayes Bigrams	alpha = 1.0, 0.5, 0.05	64
Multinomial Naive Bayes Trigrams	alpha = 1.0, 0.5, 0.05	70
Multinomial Naive Bayes Mammals	alpha = 1.0, 0.5, 0.05	30
Multinomial Naive Bayes Mammals and Trees	alpha = 1.0, 0.5, 0.05	45
Complement Naive Bayes Bigrams	alpha = 1.0, 0.5, 0.05	91
Complement Naive Bayes Trigrams	alpha = 1.0, 0.5, 0.05	88
Complement Naive Bayes Mammals	alpha = 1.0, 0.5, 0.05	18
Complement Naive Bayes Mammals and Trees	alpha = 1.0, 0.5, 0.05	36
Random Forest Bigrams	Max_depth = standard, 2, 4	70
Random Forest Trigrams	Max_depth = standard, 2, 4	66
Random Forest Mammals	Max_depth = standard, 2, 4	29

Random Forest Mammals and Trees	Max_depth = standard, 2, 4	39

Logistic Regression Bigrams	C= 1, 10, 100
	Class weight = standard, balanced, balanced
	Iters= standard, 1000, 1000
		84

Logistic Regression Trigrams	C= 1, 10, 100
	Class weight = standard, balanced, balanced
	Iters= standard, 1000, 1000
		75

Logistic Regression Mammals	C= 1, 10, 100
	Class weight = standard, balanced, balanced
	Iters= standard, 1000, 1000
		30

Logistic Regression Mammals and Trees	C= 1, 10, 100
	Class weight = standard, balanced, balanced
	Iters= standard, 1000, 1000
		41

The best performer in test was the complement naive bayes classifier for bigrams. However, the classifier that performed the most similar between test and dev with the highest results across the board was logistic regression, potentially suggesting independence in stories alongside the ability to differentiate between in-groups and out-groups. Since complement naive bays tests the likelihood of an item not belonging to a class we can look at the results of the naive bayes bigrams test and further investigate.

Default Test Complement Naive Bayes Classifier Results

	Precision	*Recall*	*F1-Score*	*Instances*
Arabic	1.00	0.86	0.92	7
Australian_Ethnic	1.00	1.00	1.00	3
Chinese	1.00	1.00	1.00	11
Indian	0.80	1.00	0.89	16
Japanese	1.00	1.00	1.00	8
Filipino	1.00	0.83	0.50	6
Russian	0.83	1.00	0.91	5
			Total accuracy: 91%	Total instances: 56

In table A we can see that almost all regions had perfect classification. Looking at a confusion matrix helped clear up some of the misclassified data, as most misclassified nations were misclassified as Indian, which could be because of the imbalanced data. However, this could be something to explore further because of potential cultural diffusion as well. Furthermore, stories were only ever mislabeled as other nations they are adjacent to. For example, Japanese stories didn't misclassify as Arabic or Australian Ethnic in either the test or the dev.

There is a marked difference in the classifiers that performed well on the ecological features versus the bigrams and trigrams, with the largest difference between the two naive bayes classifiers. None of them performed well as independent variables, but even worse as identifiable via complements. When investigating this I found that most cultures seemed to write about the same creatures, with deer, foxes, and rabbits as the top-appearing animals, crossing borders and thus making in and out-groups difficult. This is supported by what I learned while working at Glacier National Park, where my boss taught me about the common themes of bear stories and wolf stories across cultures: many stories share the same information (or disinformation) about the same animals. The major exception in the data was the Australian Ethnic class which had a total of zero tree or mammal features in train or dev.

In the future I'd like to add more stories to the database to help alleviate the issue of features not repeating. This dataset is rather small. For example, I own a book of Jewish folktales which has about 100 Chassidic Czech Ashkenazi stories in it, while the kaagle database only has four Jewish stories and doesn't differentiate between sub-ethnic units in the diaspora. I'd also like to combine the idea of extracting ecological features into more of a named entity recognition task. For example, my favorite Chinese folktale has a cowherd as a main character, but never mentions cows explicitly. If I have named entity recognition of mammals I could correctly identify Chinese stories as having cows, which my features were not able to do, but also prevent issues like ash (the burnt substance) popping up as ash trees.