Language Recognition & First Sentence Prediction

Data: 22,000 paragraphs of 22 languages

Goal 1

Predict the language of a text.

Goal 2

Predict whether a sentence is the first in its paragraph.

Language Recognition: My Approach

Create four different models

Step 1: Vectorize the text data using Count Vectorizer and Tf-idf

Step 2: Train Logistic Regression and Naive Bayes models on the data.

Step 3: Test predictive ability of each model using Internet text from the 22 languages

Language Recognition: Findings

Metrics

		Training Data Accuracy	Test Data Accuracy
Count Vectorizer	Logistic Regression	0.993750	0.933636
	Naive Bayes	0.991818	0.955682
Tf-idf	Logistic Regression	0.974489	0.954545
	Naive Bayes	0.983239	0.943409

The best model was CountVectorizer with Naive Bayes.

The model failed to predict correctly on Japanese and Chinese texts. This was due to these languages not being properly tokenized since there are no spaces between words in these languages.

Language Recognition: Ideas for Further Research

Use packages like Jieba and Nagisa to properly tokenize Chinese and Japanese.

Try using a language dictionary instead of machine learning to solve the problem.

Expand the language corpus to include more langauges.

Language Recognition: Recommendations

Incorporate the model into a translation app so that the app would know which language it should translate.

First Sentence Prediction: My Approach

Part 1: Create a new, self-supervised dataset containing the sentences and their labels for one language (Chinese).

Step 1: Split the data into training and test sets so as not to split apart paragraphs later.

Step 2: Create a function that takes a spaCy document object, splits the paragraphs into sentences, and labels each sentence as first in its paragraph or not.

Step 3: Use the function on the training and test data to create training and test dataframes.

Part 2: Random over sampling

Step 1: Because the data is highly imbalanced, use random over sampling on the minority class (first sentences).

Step 2: Do this for both training and test data.

Part 3: Latent Semantic Analysis

Step 1: Transform sentences into bag-of-words using CountVectorizer and Tf-idf.

Step 2: Use Truncated SVD with 75 components to turn the document term matrices into latent semantic analyses.

Step 3: Do this for training and test data.

Part 4: Fit a Logistic Regression model to the data

Step 1: Fit the training data to the model.

Step 2: Make predictions and evaluate the results.

Part 5: Optimize the best model (Tf-idf with Logistic Regression) for F1 score

Step 1: Create a grid search with the scoring parameter set to ‘f1.’

Step 2: Fit the training data to the model.

Step 3: Make predictions and evaluate the results.

First Sentence Prediction: Findings

Metrics

		Training Data	Test Data
Count Vectorizer	Accuracy	0.646236	0.577063
	F1 Score	0.608370	0.466718
Tf-idf	Accuracy	0.676158	0.587985
	F1 Score	0.663322	0.577999

Considering random guessing would yield about a 20% accuracy, I think this model is pretty good for a first attempt.

First Sentence Prediction: Ideas for Further Research

Build a function that can take sentences of Chinese text and predict whether they are a first sentence in a paragraph. I could do this on Wikipedia articles.

Follow the same approach but do it using English text scraped from Wikipedia.

First Sentence Prediction: Recommendations

Build a list of most common first sentences and then analyze them to find similarities.