Intelligent Systems Lab 2
Is this sentence Dutch or English?

Name: Jennifer Liu

Features

Note: Please refer to README.txt to run this program

Features for detecting English sentences

Does it contain English stop words?

In natural language processing, stop words are words that are required to be in a sentence, but does not add content to the sentence. “the” would be a prime example of a stop word in English. The NLTK library has compiled a list of stop words. I check whether the sentence contains an English stop word or not. If it contains an English stop word, it is considered English.

Most common letter combinations in English

This feature employs the same concept as feature 3. However, the letter pairings differ. For English, [“aw”, “ay”, “oy”, “kn”, “ph”] were considered. These were chosen because it will only occur in English and not in Dutch.

Does it contain an English suffix

In English, to transform a word from being a present tense to past tense, you would add an “ed” to the end of the word. Of course, there are exceptions to this rule. For instance, to represent the past tense of run, you had to use “ran”. These were known as suffixes. I scanned through every word in the sentence to see if any of these words ended with a suffix. The suffixes that were considered were ["tion", "sion", "ial", "able", "ible", "ful", "acy", "ance", "ism", "ity", "ness", "ship", "ish", "ive", "less", "ious", "ify"]. These were chosen because rarely do Dutch end with these words. The reason “ed” was not considered is because many Dutch words actually end in “ed”. Thus, making a very poor classifier.

Does it contain the English Prefix “un”

It turns out that the English Prefix “un” is one of the most common prefix. My feature looks at every word in the sentence and see if there are any words that start with “un”. If there exist such a combination, consider it English.

Most common words in English

For English, I considered the most common nouns to be seen because if I just picked the most common words, most of them were stop words, so the feature will serve the same function as the stop word feature. These were the list of most common nouns. ['area', 'book', 'business', 'case', 'child', 'company', 'country', 'day', 'eye', 'fact', 'family', 'government', 'group', 'hand', 'home', 'job', 'life', 'lot', 'man', 'money', 'month', 'mother', 'mr', 'night', 'number', 'part', 'people', 'place', 'point', 'problem', 'program', 'question', 'right', 'room', 'school', 'state', 'story', 'student', 'study', 'system', 'thing', 'time', 'water', 'way', 'week', 'woman', 'word', 'work', 'world', 'year']

Possessive Pronouns in English

In English, we often use possessive pronouns to represent that this noun/object belong to someone. For instance, “this is zara’s toy”. Using the possessive pronoun, “zara’s”, we can infer that the toy belongs to zara. It turns out that pronouns were a lot more common in English than it is in Dutch. Hence, if there was a possessive pronoun, it is more likely that it is English than it is Dutch.

Features for detecting Dutch sentences

Does it contain Dutch stop words?

This approach is very similar to checking whether an English stop word exist or not. Instead of checking for English stop words, it checks for the existence of Dutch stop words.

Most common letter combinations in Dutch

According to sources, there are certain letter combinations that frequently appear in Dutch but not in English. In my features, I considered 6 of these pairings, [“uu”, “aa”, “ieu”, “ij”, “ooi”, “oei”]. In fact, these pairings would never appear in any English words because English does not allow it.

Does it contain Dutch suffix

The Dutch suffixes that were considered were, ["ische", "thisch", "thie", "achtig", "aan", "iek", "ief", "ier", "iet", "een", "ant"].

Most common words in Dutch

With some research, I have discovered that, ['ik', 'je', 'dat', 'ze', 'hebben', 'weet', 'kan', 'ja', 'nee', 'bent', 'doen'], were the most common words to be seen in a sentence. I tried to consider words that are not stop words, so this feature does not occlude with the stop word feature.

Training and Testing Samples

In total, there were 1610, 15 word, sentences in the training sample. 713 of them were Dutch sentences, and 897 of them were English sentences.

There were 404 testing sample. 179 were Dutch sentences, and 225 were English sentences.

These samples were created using the python program data_collections.py.

I tried to keep the dataset as balanced as possible. Hence, having a near equal amount of Dutch and English sentences for the training dataset. This prevents the classifier from classifying one language better than the other.

Another factor that I considered during data collection was that I tried getting a variety of contents – News articles, novels, and textbook. My assumption is that the structure of news article would be very different when writing a novel versus writing a news article. Hence, by having a range of different types of text from both languages, it can help generalise the classifier so that it works well on a wide range of text rather just working well on news article or novels.

Where did the Data come from?

The novels were gathered from Guttenburg.org, and the news article were gathered from popular news source such as CNN, BBC. As for Dutch, these news sources were gathered from https://dutchdailynews.com/.

Creating the Validation Dataset

In order to evaluate how well my decision tree or AdaBoost was doing, I included the label inside my own test data. With that, I was able to check whether my classifier predicted the language correctly. I counted the number of correct predictions, and divided this number of samples there are. For instance, for the test data I created, if I got 300 correct, the accuracy score will be 300/404 because there were 404 sentences in total.

Decision Tree Learning Approach

For the sake of simplicity, I first converted the raw file into a binary form. For instance, if the sentence was classified as "Dutch", it is a 0, else if it is classified as "English", it is a 1. I also checked whether each feature was true or not. Consequently, I will create a file that looks something like this:

CommonDutch	CommonEng	VowelCombDutch	VowelCombEng	StopwordsDutch	StopwordsEng	EndDutch	EndEng	PrefixEng	PossessivePronoun	Lang
1	0	0	0	1	0	1	0	1	1	1

For each attribute, if it existed, it is a 1, otherwise it is a 0.

Using this parsed file, I find the best possible attribute to split on using information gain. In the example above, I first split the data by asking whether a common Dutch word existed. If the sentence contained a Dutch word, it goes on the right branch of the tree, otherwise it goes to the left branch of the tree. I calculated the information gain by finding the difference between the entropy of parent node and the entropy of the current split. I take the attribute that gave us the lowest information gain and create a tree node out of it.

The structure of the tree node saves 3 things, the best attribute, left branch, and right branch. We recurse on the right and left branch to find the best split given the remaining data on the left and right branch.

Evaluation Result Summary on Test Data

Upon trying different depths, we discover that depth 2 was the best one to choose. Trees that had a depth greater than 2 was not performing any better, but not any worse either.

	Depth = 1	Depth = 2	Depth = 3	Depth = 4
Accuracy	98%	99%	99%	99%

Best Decision Tree Classifier:

if VowelCombDutch <= 0:

```
  if CommonDutch <= 0:
        Lang = 0
  Else:
        Lang = 0
Else:
    If StopwordsEng <= 0:
        Lang = 0
    Else:
        Lang = 1
```

Boosting Learning Process

When using 1 stump for the Boosting learning process, it reached a maximum accuracy of 98%. However, as I considered more ensembles, the accuracy started reducing. In fact, when all 10 of decision stumps were used, I achieved an accuracy score of 60%. I suspect that the reason this is the case is because only a few of my attributes were helpful. It is notable that with 1 best stump, it already reached an accuracy of 98%. The other stumps were not very helpful. For instance, when using common English words as one of the decision stump, it gave me an accuracy of 44%. This is less than random guessing. Clearly, this was not a very helpful feature. Due to this fact, I believe this is the reason why combining many weak learners was more detrimental than just choosing the best stumps to use.