Foundations of Intelligent Systems

Lab 2

Suhas Cholleti

List of features:

1. English common words : I have taken the 1000 most common words used in english. Studies show that 1000 most common words are 76% of all non fiction literature and 88% of oral communication. I have tried with 10 most common words too, but taking 1000 didn't hurt the performance and improved the accuracy. Removed the words that are common in both English and Dutch common words. Added a few words from the English Stop Words.
2. Dutch Common words: I have taken the 1000 most commonly used dutch words for similar reasons as above. Removed the words that are common in both English and Dutch common words. Added a few words from the Dutch Stop Words.
3. English prefix: Added a few common english prefixes. A lot of the words in english words start with im-, un-, dis-, ir-. If there are words starting with these prefixes, there is a good chance that these words are english. Used only 4 of them are 97% of all the prefixed words in english start with these four prefixes.
4. Dutch prefix: Added a few common dutch prefixes. A lot of the words in dutch words start with ge-, be-, her-, etc. If there are words starting with these prefixes, there is a good chance that these words are dutch.
5. Dutch suffix: Added a few common dutch suffixes. A lot of the words in dutch words ends with -ische, -thisch, -thie etc. If there are words ending with these suffixes, there is a good chance that these words are dutch.
6. English suffix: Added a few common english suffixes. A lot of the words in english words end with -sion, -tion, -ial, -ful etc. If there are words starting with these suffixes, there is a good chance that these words are english.
7. Letter combinations: There are 16 vowel combinations that are commonly used in Dutch. If this letter combination exists in a word, we can safely predict that the word could be ducth.
8. Number of english words > number of dutch words : Using the above six features, we count the number of words that algorithm thinks are english and dutch. As the individual features might go wrong, i used this as this effectively combined the above features into 1 feature.

9. Average length of the word : The average length of a word in english is 8.23 characters and the average length of a word in dutch in 9.70 characters. I am checking if the average length of the 15 words is less than or greater than 9. As each example only has 15 words, this could predict wrong as all words could be chosen which are large/small. If the length of each example is higher, this would have been a more useful feature.
10. Does letter q exist in the sentence:

Decision Tree : I have used an entropy calculation to select the column to use at each node. The column with the least entropy has the highest information gain. At the start of each recursion, the minimum entropy value is selected and the feature list is partitioned on that column into 2 parts, one having all the true rows and the other having false rows. The minimum entropy column becomes the new node and 2 recursive calls are made with the true and false rows for the left and right child on the node.

There are 3 base conditions for the above recursion. If all the features belong to the same goal state, a Leafnode is returned with that goal state. If all the columns are already used, then we return the Leafnode with the goal state that most frequently occuring in the list of features. If the list of features is empty, then we return the Leafnode with the goal state that most frequently occuring in the parents feature list.

Initially I used a training set of 120 examples. That gave me a tree of height 2(including the leaves) with the root node being the feature 8 in the above list. Then changed the training set to around 1900 examples which had a few more complicated examples like “en|Vuillemins most important works include his detailed highly decorative large format Atlas Illustré de Géographie” and “en|On September she married Prince Nikolaus II Esterházy de Galantha who in became the Prince”. In both these examples there is the word “de” which means “the” in english. This is a common dutch word which I use in feature 1. Due to examples like this, I got a tree of length 10, using all the features available.

Adaboost: I found that having either 3 or 4 stumps gave me the best result.

I have used the same minimum entropy column as above to choose the column to use for the stump at each stage. Once we choose the column, we find the total error by adding the weight of all the values classified wrong. We use the error to find the significance which would be associated with each of the stumps. Once we find the significance, we calculate the new weights, increasing the weights of examples classified wrong and decreasing the weights of the examples that are classified correctly and normalize them so that they all add up to 1.

We repeat the above process to get the desired number of stumps. For me I found that 3-4 worked best. I used the same 1900 odd examples i used for Decision tree for adaboost too.

Documentation on how to use the code:

```
train.py <examples> <hypothesis_output> <learning_type>
```

Examples : Training dataset
Hypothesis_output : file to store the hypothesis
Learning type : dt for decision tree and ada for adaboost.

```
predict <testing_examples> <hyposthesis_input>
```

Testing examples : testing dataset
Hyposthesis_input : input file containing the hypothesis(pickle)