Project 1 – Naive Bayes Language Classification

Due Apr 07 in class

Contact

Instructor:	John Pate
Office:	610 Baldy
Email:	johnpate@buffalo.edu

Course

Lectures:	M/W/F 09:00 - 09:50 115 Talbert
Office hours:	F 14:00 – 15:00 R 14:00 – 15:00 224 Clemens
Course website:	UBLearns

This project is worth 25 total points.

In this project, you will implement a Naive Bayes model for language identification. I have provided data and a skeleton implementation at:

www.acsu.buffalo.edu/~johnpate/lin567_p1.zip

The skeleton implementation is in Scala. You are free to use another programming language if your implementation follows these rules:

- It must read the input files exactly in the format as distributed;
- It must provide output in exactly the same format as the Scala skeleton;
- It can be run on the command line on lin-remote.caset.buffalo.edu;
- It accepts the following command line arguments in the following order:
  1. `path_to_training_file`
  2. `path_to_testing_file`
  3. `n` (for specifying character *n*-gram length)
  4. `lambda`

If you are not using Scala or Java, please check that your programming language of choice can be run on lin-remote *before* doing your implementation. If it cannot, talk with me about possibly having it installed. Also ensure that your programming language handles unicode properly.

If you use the provided Scala skeleton, you will be able to compile with `sbt package` as in Homework 1.

The data is lines of movie subtitles from 21 different languages. Each "document" is one line from a movie, and the training file has one document per line in a pipe-separated format. The first field is an id-string, the second field is the text of the line, and the third field is the correct language (notice this is empty for the test set).

Your model will be a *character* *n*-gram model – each feature should be the number of times each sequence of *n* characters occurs. You should implement add-lambda smoothing, and `src/main/scala/io.scala` provides a method for extracting these features. If you follow the Scala skeleton, you should be able to produce output for a character trigram model with an add-lambda parameter of 1 by using the following command:

```
scala -cp path_to_jar lin567_p1.Run train.labeled dev.labeled 1 3
```

Note that you will have to use the actual path to the compiled jar. I have provided the perl script I will use for evaluation in `evaluate.pl`. To evaluate the above model, use:

```
scala -cp path_to_jar lin567_p1.Run train.labeled dev.labeled 3 1 | ./evaluate.pl -gold dev.labeled
```

The script provides classification accuracy across all languages and for each individual language.

E-mail me your project code by the due date (March 23rd in class), tell me what value for n and λ I should use on the test set, and provide a short (5-page) write-up with the following sections.

- Introduction: a paragraph that briefly explains the task of language identification.
- Model: define the Naive Bayes model you have implemented in prose and equations.
- Results:
  1. How does increasing n affect the performance of this classifier? Why might this be?
  2. How does increasing λ affect the performance of this classifier? Why might this be?
  3. How does performance vary across languages? Why might this be?