Assignment 5
CSEP 517: Natural Language Processing

University of Washington

Due: May 28, 2017

In this assignment, you will get acquainted with basic concepts from machine translation. Instead of actually building a system that does machine translation (as you will learn in class, this can be quite hard), you will build a classifier that can tell whether a translation was created by a human or by a machine.

The data for this task can be downloaded from Canvas. The data are divided into two files: a training set and a test set. The training set (`A5.train.labeled`) is formatted so that it includes a source sentence in Chinese, a human translation of that sentence to English (called the *reference*), another translation to English (either by a machine or human, called the *candidate*), and a score for the quality of the translation.[1] The score is a very simple version of Bleu that considers only unigrams. The training set also includes an additional line which indicates whether the candidate comes from a machine (`M`) or human (`H`). The test set (`A5.test.unlabeled`) has the same format, only with the value of `?` on that line.

Note that the data are encoded in UTF-8, since they include Chinese characters. You will probably want to use your programming language's support for dealing with UTF-8 text in this assignment.

Your task is to build a classifier that tells whether a candidate is a human or machine translation. In addition to the candidate, your classifier may consider the source sentence and the reference. The Bleu scores provided may also be used as inputs to your classifier, as well as more complex versions of Bleu or other MT evaluation scores you choose to calculate. You are welcome to use any existing resources, tools, or libraries to build your classifier. You may even use additional data, with the exception of the test data.

Deliverables

- Submit your predictions on the test set (detailed instructions below).
- The official evaluation score will be the average of F_1 for the human and machine classes.[2] Out of the 100 points possible on this assignment, 10 points will be awarded for beating 0.5 (approximately random), and 10 points will be awarded for beating 0.65. The top performers in the class will receive bonus points.
- Explain how your classifier works, including the resources and algorithms you used, and any procedures you used to estimate its performance while making empirical design decisions. Be sure to properly cite any existing tools or libraries.

Submission Instructions

Submit a single gzipped tarfile (`A5.tgz`) on Canvas.

[1] The script for computing Bleu scores can be found on the NIST website. You will need to reformat the data if you want to run this script.

[2] You should be able to see why a most-frequent-class classifier won't do well on this score.

- **Code**: You will submit your code together with a neatly written README file to instruct how to run your code with different settings. We assume that you always follow good practice of coding (commenting, structuring), and these factors are not central to your grade.
- **Labels**: Submit a file named `A5.test.predicted` that is in the same format as `A5.test.unlabeled` with each `?` replaced by your prediction (`M` for machine, `H` for human). The total number of lines should be exactly the same as in `A5.test.unlabeled`.
- **Report** (use the filename `A5.pdf` and include in the tarfile): Your writeup should be two to three pages long, or less, in pdf (one-inch margins, reasonable font sizes, preferably LaTeX-typeset). Part of the training we aim to give you in this class includes practice with technical writing. Organize your report as neatly as possible, and articulate your thoughts as clearly as possible. We prefer quality over quantity. Do not flood the report with tangential information such as low-level documentation of your code that belongs in code comments or the README.