Course Project
CSE 517: Natural Language Processing

University of Washington

Spring 2018
Dry run: May 9, 2018, 1 pm
Due: June 1, 2018, 1 pm
Updates: April 19, 2018
Updates: May 2, 2018

Your project is almost the same as the first assignment. Please refer to that document again, and then note the updates and clarifications below.

Bigger vocabulary. In this project, the alphabet ($\mathcal{V}$) is the set of all valid UTF-8 encodings of Unicode version 10.0 together with the 65 control characters (i.e., V = 136,755). You can read about Unicode at `http://unicode.org/`. Note that this is a significantly larger alphabet than you had in the first assignment.

Teams. This project is meant to be completed in teams of three: one CSE Ph.D. student expecting to do a Ph.D. on an NLP topic, one CSE Ph.D. student expecting to do a Ph.D. on a different topic, and one student who's not doing a Ph.D. in CSE. We recognize that it may not be possible to build a team this way, but we encourage you to try.

Follow the spec! As with the assignment, your system should only print anything in response to receiving a command. For example, it should not output instructions or other messages upon starting. This, and any other deviation from the spec given in the assignment, will break our grading scripts and you won't receive points. Again, **it is your responsibility to make sure your submission runs exactly to the specification; we will not troubleshoot your code, and any problems will result in a major loss of points.**

Speed. To ensure that your code runs fast enough for us to grade all teams' systems, an even mixture of the different commands should be processed at a rate of at least 10 commands per second.

Third goal. In addition to the two goals described in the assignment 1 document (assigning high probability to previously unseen natural text and generating text that looks natural), you have a third goal. We will use your model to classify texts as "naturally occurring" or not. The naturally occurring texts could be any text in any natural language. The "unnatural" texts will come from other teams' language models. Your model will be used to rank a collection (half natural, half "unnatural") by perplexity scores, and we will measure how well your model separates natural from unnatural texts. (Showing that $p(c_{1:I} \mid natural)$ is monotonic in $p(natural \mid c_{1:I})$, under reasonable assumptions, is left as an exercise.)

Regarding the second goal. As before, we will evaluate your program based on its ability to convince *other* teams' programs that it is, in fact, natural. You may be tempted to try to generate memorized data. If you do this, your score will suffer if we decide to mix up `o` and `g` commands!

Optional dry run. You are invited to submit a version of your system as a dry run (see deadline above). We'll execute the dry run evaluation the same way as the final evaluation. Participation in the dry run is optional, but please note:

- If your team does participate in the dry run and everything executes flawlessly with no extra effort by the course staff, then your project grade will be 40% satisfied. That is, the final system will count for 60% of the project grade. We will be more generous in grading dry run systems and will give feedback and advice based on the writeup. (To be precise, if d is your dry run system score and f is your final system score, then your final project grade will be $\max(f, 0.4 \times d + 0.6 \times f)$.
- If your team does *not* participate in the dry run, the entire project grade will be based on the final system.

Submission. Each submission should follow the instructions for assignment 1, but name the gzipped tarball `project.tgz`. Canvas will allow you to set up your team as a group and submit once for the group. Include in your submission a file with the name `README`, as before. You may write a few paragraphs; be complete in your description, but concise. We expect your model to incorporate intuitions about natural language that you learned from the course, and support your design choices on model and data with experiments.

Outside software. We have no ban on software you can package with your submission. The only restrictions are those imposed by `umnak.cs.washington.edu`, the server we'll be running your system on. We will allow five minutes from the time we run the bash script in your submission until your system is required to start responding to the queries, which will allow for a little set-up. Please also note that you must assume we will not have a connection to the internet when running your submission, so having your code download and install packages is a bad idea.

Grade. While your grade is mostly determined by the correctness of your implementation—is it a proper language model? do you exactly follow the spec?—the clarity of your writeup is also important. The course staff are happy to give feedback on drafts submitted sufficiently in advance of the project deadline (the earlier the better). Bonus points may be awarded for top-performing teams in terms of perplexity on natural data (which should be low), perplexity of other teams' systems on your generated data (which should be high), and classification accuracy in the natural-or-not evaluation described above.