Course Project

CSE 447: Natural Language Processing – University of Washington

Winter 2022

The year is 2042. As a result of many years of peace and cooperation, the international community has invested in a massive expansion of the International Space Station. On a given day, hundreds of people are in orbit. Sophisticated tools have been developed to enable high speed, private communication among individuals, and also between the station and Earth. You have been hired to create a system that will allow astronauts to send natural language messages without talking, using advanced eye-tracking hardware. (The interface for presenting incoming messages to the astronaut is not your responsibility)

The system works by supporting, at each time step, the astronaut's choice of the next character in the message sequence. The system displays three Unicode characters in the astronaut's visual field; they select one, or, if the next character they need is not among the three, use a more complicated (and slow) secondary procedure to choose another option. Naturally, communication will be faster (and require less work on the part of the astronaut) if the system can accurately predict the next character of the intended message and place it among the top three. (The implementation of the secondary selection procedure is not your responsibility.)

Overall, your system's goal is to save astronaut time. There are two parts to this goal, which may be in opposition to each other. You should aim to minimize:

- Processing time: after the astronaut chooses the ith character, your system should choose the top three candidates for the $(i + 1)$th character as quickly as it can.
- Error: as often as possible, the astronaut's next choice should be among the three candidates. We will count the fraction of times it is not in the top three and refer to it as the error rate.

In the simulation test at the end of the quarter, we will measure both the processing speed and accuracy of your system on real data sent by real simulated astronauts.

Additional notes:

1. The astronaut will speak at least one human language, but you don't know which one(s). The secondary selection procedure allows the astronaut to choose any Unicode character; you can imagine that the cost of navigating through it will be very high. The decision of which Unicode characters to allow to be possible candidates is up to you; if you are too restricted, your system might have more errors on messages in some languages. If you try to allow all Unicode characters at every time step, your processing time might suffer.
2. The specification for your program is quite simple, alternating between two steps:
    - At the start of an iteration (including the beginning of execution), your program outputs three characters, representing a prediction that the astronaut will choose one of them.
    - The program waits until the astronaut enters the next character of their message, which will arrive on standard input. For the purposes of measuring processing time, the clock starts when this character is entered and stops when the three characters arrive.

3. How you build your system is entirely up to you. You are free to use any resources (code or data) that are available to you. Obviously, you must not violate any laws or terms of service.

4. The project is meant to be completed by teams of three people.

Deliverables and Deadlines

At each checkpoint, you will turn in all source code and an executable program or script. Please sign up for your groups and submit your checkpoints on Canvas. When submitting your project, please follow the instructions and specifications in this GitHub repo. The checkpoints are as follows; the **deadlines for each checkpoint are shown on the course calendar**.

1. **Around week 4**. We'll check that your program runs to spec (no error or processing speed measurements). You're graded only on turning in the program on time and running to spec. You must also submit a short document that contains the following:

   - Dataset: what kind of data are you going to use to train your model, and how will you obtain this data?
   - Method: what kind of method will you use, and how will you implement it (e.g. language, framework)?

2. **Around week 6**. We'll check that your program runs to spec, and we'll measure error and processing time (and report them back to you along with your rank among teams in the class on both measurements). You're graded only on turning in the program on time and running to spec, but your team will get a bonus point if your system is on the Pareto frontier (i.e., there may be systems with lower error or lower processing time than yours, but no system with both).

3. **Around week 8**. Exactly like checkpoint 2, but for full credit you must show improvement over your checkpoint 2 measurements (either a reduction in error rate or a reduction in processing time, or both).

4. **Around week 10**. Final deadline. The course staff will give up to three bonus points to systems on the Pareto frontier (i.e., there may be systems with lower error or lower processing time than yours, but no system with both).

At checkpoint 4, you will also turn in a report of your project. Please ensure:

- The report must be no more than one page (references don't count against the page, figures/tables do), letter size, 1-inch margins, 11-point Times font, submitted as a pdf.
- Describe your approach, making use of concepts and methods learned in class.
- Describe the data you collected, existing datasets you used, and existing code libraries or packages you used. If there's any question at all about how to acknowledge the work of others, talk to the course staff and we'll be happy to help.

The course staff may offer bonus points to exceptionally well-written reports. The course staff will also award up to three bonus points for well written data statements. Please see instructions at the end of the document.

Grades

By default, all members of your team will share the same grade, counted out of 50 points.

- Checkpoint 1: 5 points
- Checkpoint 2: 5 points (1 bonus point possible)
- Checkpoint 3: 5 points (1 bonus point possible)
- Checkpoint 4 (final test): 15 points (3 bonus points possible)
- Final writeup: 20 points (6 bonus points possible)

Individual Report

Students in this course are expected to work together professionally, overcoming the inevitable challenges that arise in the course of a team project. We recognize that, occasionally, team members behave unreasonably. To help us navigate situations where you feel a shared grade would be unfair, we invite you to submit individual updates on your team's progress at any time during the quarter using this form.

Data Statement

For extra credit in the final checkpoint, your team may submit a Data Statement [1], which will require you to examine your data sources. Your data statement will consist of a document, which answers the following questions. We have included examples in each question below to guide your responses. If you are unable to measure a particular characteristic of your dataset, clearly state so (and provide justification of why it is difficult), and suggest possible way(s) you could measure that characteristic, if you had infinite resources. We will award bonus points based on effort.

Data source What is the source of your data? (dataset from X, crawled from website Y, curated by research group Z, machine generated, etc.)

Data size What is the (total) size of your data? What is the data size for each language?

Preprocessing procedure How is the data preprocessed? Please also describe any steps of data cleaning/filtering.

Curation rationale Which texts were included and what were the goals in selecting these texts (for each language)? For example, you may write something like "we included books from genres A, B, and C because we believe they will cover a large set of vocabulary."

Language variety What languages do your texts include? Give English descriptions along with ISO 639-1 language codes for each language in your texts. For example, if your text contains English and German, you would write: "English (en), German (de)."

Speaker demographics Who produced the texts in your data? Include as much information as you can for each of these items: age, gender, race/ethnicity, native language , socioeconomic status, number of speakers. For example, if your data is crawled from Twitter, you should do your best to ascertain the age, gender, and language distributions for that platform's users (and of course, cite your sources).

Speech situation In what situations are the texts being produced? Details may include: modality (spoken/signed, written), scripted/edited vs. spontaneous, intended audience.

Text characteristics What are the genres and topics of the texts? For example, you may write something like "scientific fiction books," "biomedicine journal articles," "comedy movie scripts," etc.

Ethical considerations What ethical concerns are associated with your dataset and/or curation procedure? For example, you may mention that you did not receive consent from users to use their social media posts, but that you have curated the data in accordance with the platform's terms of service (be sure to check!).

References

[1] Emily M. Bender and Batya Friedman. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604, 2018. doi: 10.1162/tacl_a_00041. URL https://aclanthology.org/Q18-1041.