Отчет

Структура проекта

После завершения первой части домашки я пошла переносить все в reflex-овский проект в PyCharm. По канонам проектирования приложений я сделала несколько файликов, каждый из которых отвечает за свою логическую часть:

- “corpus_collection.py” содержит весь код, который касается предобработки текста и получения из него нужного корпуса данных

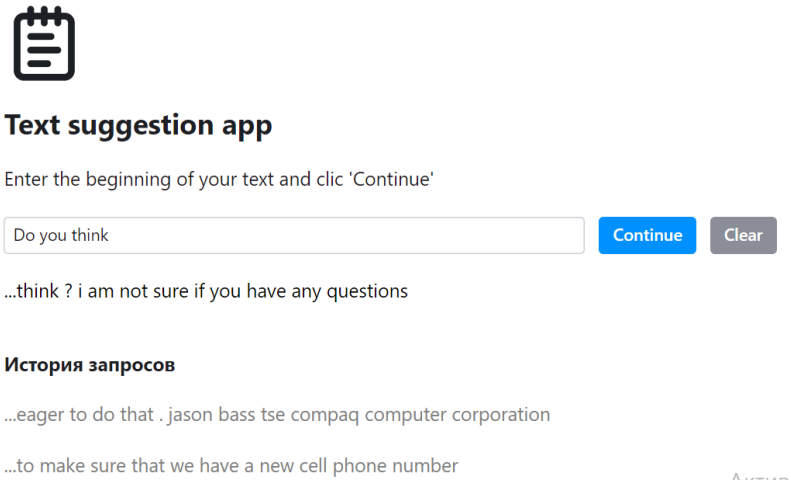
- “text_suggestion.py” содержит все классы, относящиеся к предсказанию текста

- “reflex_projext.py” это основной файл, содержащий интерфейс reflex-а

В ходе работы с большим корпусов в коде функций я выявила несколько косяков, поэтому добавила несколько проверок и обработок corner case-ов (предложения в корпусе длины < n, ненайденные префиксы и тд). В остальном код не особо поменялся.

Фронтенд

Что касается фронтенда, то reflex вообще очень долго не хотел у меня нормально устанавливаться и заводиться. В итоге все заработало, и я сделала достаточно простую интуитивно понятую страницу. Поисковая строка, кнопка поиска, хранение истории запросов глубины 3, кнопка очищения истории. И даже иконка с блокнотом☺



Данные

Предобработку писем я сделала в первом задании домашки, в целом все комменты в коде есть, еще раз комментировать не буду. Я сохранила все “emails_processed.csv”, чтобы не тратить на обработку текста время при запуске приложения, и пользовалась функцией extract_corpus, которая просто достаёт обработанные данные и сплитит по словам. Но для других задач я все же оставила возможность обработки прям в приложении (функции clear_text и collect_corpus).

TO DO

Очень жалко, что дана всего неделя на эту домашку, я бы еще над ней посидела и сделала прям красиво. Во-первых, поковырялась бы в данных и получше их почистила. Там много разных типов писем, разные corner-case, которые было бы более качественно обработать.

Во-вторых, наверняка есть какие-то приколы, как улучшить алгоритм генерации. Например что-то делать с несуществующими префиксами, чтобы они обрабатывались. Как-то научиться брать больший контекст (например n=10), чтобы при этом можно было продолжать фразу длины <10. Но это надо копаться и изучать, а времени не хватило на это.

Ну и интерфейс конечно можно было бы красивый запилить, чтобы прям как ChatGPT, какой-нибудь доп функционал и стиль страницы красивый.