Advanced ML: Домашнее задание 4

Четвёртое домашнее задание посвящено достаточно простой, но, надеюсь, интересной задаче, в которой потребуется творчески применить методы сэмплирования. Как и раньше, в качестве решения **ожидается ссылка на jupyter-ноутбук на вашем github (или публичный, или с доступом для snikolenko); ссылку обязательно нужно прислать в виде сданного домашнего задания на портале Академии**. Как всегда, любые комментарии, новые идеи и рассуждения на тему категорически приветствуются.

В этом небольшом домашнем задании мы **попробуем улучшить метод Шерлока Холмса**. Как известно, в рассказе *The Adventure of the Dancing Men* великий сыщик расшифровал загадочные письмена, которые выглядели примерно так:

Пользовался он для этого так называемым частотным методом: смотрел, какие буквы чаще встречаются в зашифрованных текстах, и пытался подставить буквы в соответствии с частотной таблицей: E — самая частая и так далее.

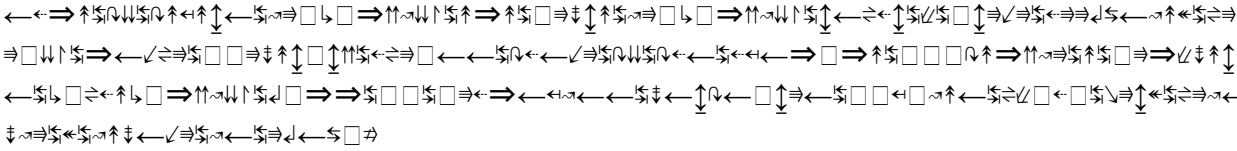
В этом задании мы будем разрабатывать более современный и продвинутый вариант такого частотного метода. В качестве корпусов текстов для подсчётов частот можете взять что угодно, но для удобства вот вам “Война и мир” по-русски и по-английски:

https://www.dropbox.com/s/k23enjvr3fb40o5/corpora.zip

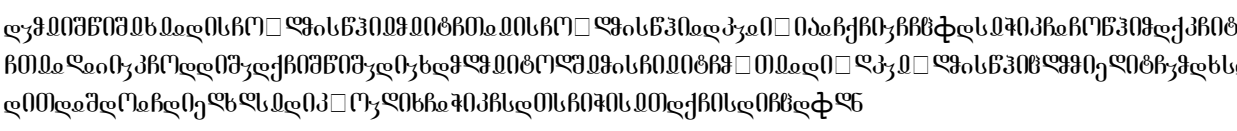
1. Реализуйте базовый частотный метод по Шерлоку Холмсу:
    - подсчитайте частоты букв по корпусам (пунктуацию и капитализацию можно просто опустить, а вот пробелы лучше оставить);
    - возьмите какие-нибудь тестовые тексты (нужно взять по меньшей мере 2-3 предложения, иначе совсем вряд ли сработает), зашифруйте их посредством случайной перестановки символов;
    - расшифруйте их таким частотным методом.
2. Вряд ли в результате получилась такая уж хорошая расшифровка, разве что если вы брали в качестве тестовых данных целые рассказы. Но и Шерлок Холмс был не так уж прост: после буквы E, которая действительно выделяется частотой, дальше он анализировал уже конкретные слова и пытался угадать, какими они могли бы быть. Я не знаю, как запрограммировать такой интуитивный анализ, так что давайте просто сделаем следующий логический шаг:
    - подсчитайте частоты *биграмм* (т.е. пар последовательных букв) по корпусам;

- проведите тестирование аналогично п.1, но при помощи биграмм[1].

3. Но и это ещё не всё: биграммы скорее всего тоже далеко не всегда работают. Основная часть задания — в том, как можно их улучшить:
   - предложите метод обучения перестановки символов в этом задании, основанный на MCMC-сэмплировании, но по-прежнему работающий на основе статистики биграмм;
   - реализуйте и протестируйте его, убедитесь, что результаты улучшились.
4. Расшифруйте сообщение:



Или это (они одинаковые, но сообщали о проблемах с юникодом):



5. *Бонус*: а что если от биграмм перейти к триграммам (тройкам букв) или даже больше? Улучшатся ли результаты? Когда улучшатся, а когда нет? Чтобы ответить на этот вопрос эмпирически, уже может понадобиться погенерировать много тестовых перестановок и последить за метриками, глазами может быть и не видно.
6. *Бонус*: какие вы можете придумать применения для этой модели? Пляшущие человечки ведь не так часто встречаются в жизни (хотя встречаются! и это самое потрясающее во всей этой истории, но об этом я расскажу потом).

[1] В качестве естественной метрики качества можно взять долю правильно расшифрованных букв или, если хочется совсем математически изощриться, расстояние между двумя перестановками, правильной и полученной из модели; но, честно говоря, в этом задании следить за численными метриками не так уж обязательно, будет и глазами всё видно.