Проект этап 2. Разбиение набора данных на биграммы

Чтобы изучить, как устроены тексты, мы обычно должны принимать во внимание не только отдельные слова, но и последовательности слов и отношения между ними. Эти последовательности могут состоять из любого количества слов, но обычно их количество ограничено двумя или тремя.

Последовательность любого количества смежных токенов называется **n-граммой**, где **n — количество токенов**. Последовательность двух соседних токенов называется **биграммой**. Неудивительно, что последовательность из трех соседних токенов представляет собой **триграмму**.

На данный момент мы будем придерживаться только **биграмм**.

Говоря о биграммах в этом проекте, мы делим их на две части: **голову и хвост**. В нашем случае **первый токен биграммы — это голова**, а **второй токен — хвост**. Например, в биграмме **good night**, good - голова, a night - хвост.

Описание

После того, как обучающие данные получены и предварительно обработаны, их необходимо преобразовать в **модель цепи Маркова**. Первый шаг — сопоставить связи между токенами в корпусе. Для этого мы будем использовать биграммы.

Задачи

1. Преобразовать корпус в набор биграмм. Результаты должны содержать все возможные биграммы из корпуса, а это значит, что:

— Каждая лексема из корпуса должна быть головой биграммы, за исключением последней лексемы, которая не может стать головой, так как за ней ничего не следует;
— Каждая лексема из корпуса должна быть хвостом биграммы, за исключением первой лексемы, которая никак не может быть хвостом биграммы, потому что ей ничего не предшествует. 2. Выведите количество всех биграмм в корпусе. 3. Возьмите целое число в качестве пользовательского ввода и напечатайте биграммы с соответствующим индексом. Повторяйте этот процесс до тех пор, пока не будет введена строка `exit`. Кроме того, убедитесь, что ввод на самом деле является целым числом, попадающим в диапазон набора биграмм. Если это не так, распечатайте сообщение об ошибке и запросите новый ввод. Сообщения об ошибках должны содержать типы ошибок (`Type Error` , `Value Error` , `Index Error` и т.д.). Каждая биграмма должна иметь формат Голова: `[head]` Хвост: `[tail]` и печататься с новой строки.

Вы должны печатать только выходные данные текущего этапа, а не предыдущего, но, как и на предыдущем этапе, имя файла, содержащего корпус, должно быть указано в качестве пользовательского ввода.

Пример

Символ «больше чем», за которым следует пробел **(>)**, представляет ввод пользователя. Вот как должен выглядеть ожидаемый результат. Табуляции и пробелы не имеют значения во время тестирования, но символы новой строки имеют значение.

```
> corpus.txt
Количество биграмм: 2343554

> 0
Голова: What     Хвост: do
> 4
Голова: They're  Хвост: savages.
> 5
Голова: savages. Хвост: One
> 34
Голова: I've     Хвост: never
> 42
Голова: ever     Хвост: in
> 256
Голова: the      Хвост: lads
> 453
Голова: sentence Хвост: you
> 2345
Голова: don't    Хвост: understand
> 3000
Голова: can      Хвост: protect
> 943287563823572346
Index Error. Пожалуйста введите число, которое не больше количеста
биграмм.
> six
Type Error. Пожалуйста в введите число.
> -1
Head: the      Tail: North!
> exit
```

Совет

`nltk.bigrams` может вам помочь. Не забудьте преобразовать его в правильный тип и не забудьте о форматировании в конце.