Генератор текста

Задача состоит в том, чтобы написать автоматический генератор текстов с помощью цепей Маркова (http://ru.wikipedia.org/wiki/Цепь_Маркова).
Примерный алгоритм работы программы заключается в следующем. На стадии обучения программа должна для каждого слова *w1* посчитать, с какой частотой произвольное другое слово *w2* встречается после слова *w1*. Далее, на стадии генерации текста, программа должна после очередного слова генерировать следующее в соответствии с посчитанным распределением частот. Аналогично можно генерировать очередное слово в зависимости от двух предыдущих и так далее.

Предложения и знаки препинания должны быть максимально похожи на нормальные тексты: предложение начинается с заглавный буквы, пробел стоит только после знака препинания, а не до него, не встречается несколько знаков препинания подряд и т.п.

Детали реализации

Первый этап работы программы — токенизация текста, то есть разбиение его на составные части (токены), которые далее будут рассматриваться как «слова». Токеном будем называть последовательности символов, целиком состоящие из букв, последовательности символов, целиком состоящие из цифр, и отдельные символы. При разбиении текста на токены нужно брать максимально длинные токены из возможных (то есть "abc" нужно рассматривать как один токен, а не как "a", "b", "c").
Далее, программа должна уметь подсчитывать частоты слов после заданной цепочки слов. При этом у программы есть параметр *D* — максимальная глубина, — и программа должна подсчитывать частоты для всех последовательностей не длинее *D*. Это нужно для того, чтобы при генерации была возможность стабильно генерировать начала предложений, когда сгенерированная история еще короткая. Статистику нужно подсчитать в том числе и для пустой истории (это по сути просто частоты слов).

Для программы также нужно написать юнит-тесты (в том же файле). Юнит-тесты должны проверять работу всех содержательных частей программы.