Segundo Exercício-Programa: Gerador de Poesia Pseudo-Dadaista

BCC 2018 - MAC0110 - Entrega: até 13/05/2018 23:55 pelo PACA

Introdução

Neste segundo exercício-programa, você irá criar um programa que compõe automaticamente pérolas da literatura universal, como as que seguem:

Exemplo 1	*Exemplo 2*
O céu gosta da beterraba *como o inverno pensa na lasanha.* *O garfo come a ideia* *porque o pássaro da diarreia ri.* *Passa pelo amor a água* *toda vez que a árvore sai do frango.* *Enquanto o mosquito a enchente lambe* *o calor entra no dicionário.*	*Oculta o asno a cerveja* *se o ferro na nebulosa rasteja.* *Como ri da chacrete a besta* *e toca o fanfarrão o planeta.* *Quando o coelho a fogueira assa* *o joelho pelo fraco passa.* *Usurpa a diarreia a fome* *se a viola o bagulho come.* *A ordem teme a ideia* *mesmo quando o micróbio o zoológico desafia.*

Notem a riqueza vocabular e a multiplicidade de imagens e significados. Observem as inversões de frases e as rimas ricas no segundo exemplo. Quantas sutilezas! Que perspicácia!

Os segredos do ofício

Para a produção de tão plurissignificativas preciosidades, o computador vai precisar de algum material de onde partir. Parte deste material será fornecido pelo usuário, enquanto outra parte fará parte do próprio programa. Veremos todos os detalhes em detalhe.

Para simplificar nossa vida, restringiremos um pouco o universo linguístico do nosso "poeta automático". Ele utilizará apenas substantivos no singular e verbos transitivos diretos ou indiretos, na 3a pessoa do singular, no modo indicativo. Assim estão excluídos pronomes pessoais, verbos intransitivos, verbos transitivos direto e indireto, verbos reflexivos, voz passiva, subjuntivo, etc., etc., etc.

Especificamente, o usuário deverá fornecer ao computador uma lista de substantivos no singular, juntamente com o respectivo artigo, como "a árvore", "o morcego" ou "a imensidão", e também uma lista de verbos, conjugados na 3a pessoa do singular, juntamente com uma preposição (no caso dos verbos transitivos indiretos), como "passa por", "pensa em" ou "machuca -" (o hífen simboliza a ausência de preposição, pois trata-se de um verbo transitivo direto).

O usuário também vai informar se deseja ou não utilizar rimas, e quantos versos o poema deverá ter. Seu programa poderá gerar poemas tão longos quanto Os Lusíadas e tão concentrados quanto um haikai:

> *A goiabada com o cão viaja*
>
> *a nuvem ofusca a inveja.*
>
> *A risada trafega pelo pêssego.*

Nos exemplos de poemas fornecidos na primeira página, podemos observar as principais características das frases:

- elementos: um sujeito, um verbo e um objeto (direto ou indireto);
- inversões: sujeito+verbo+objeto ou sujeito+objeto+verbo ou verbo+objeto+sujeito;
- conjunções: como, e, enquanto, mesmo quando, porque, quando, se, toda vez que;
- rimas: no segundo exemplo, os finais dos versos pares "rimam" com o final do verso anterior.

Na seção "Implementação" são fornecidos mais detalhes sobre estas características e como o seu programa pode utilizar as palavras fornecidas pelo usuário na geração de poemas.

Implementação

Neste EP você deverá fazer um programa que peça ao usuário uma lista de substantivos e uma lista de verbos, e permita ao usuário gerar poemas pseudo-dadaístas com ou sem rima. Para isso você deverá criar um programa em Python chamado `gerador_dadaista.py`, que possui *obrigatoriamente* as seguintes funções:

- `def produzVersos(substantivos, verbos, númeroDeVersos, rima)`: gera (na tela) um poema, usando as listas de substantivos e verbos e com o número de versos desejado, com ou sem rima, dependendo do valor do parâmetro booleano `rima`.
- `def main()`: programa principal, que lê as listas de palavras, pergunta se o poema deve ou não usar rimas e qual o número de versos desejados, e gera o poema.

Usaremos nossos programas tanto em modo interativo (rodando-o no ambiente Idle ou no terminal com `python3 gerador_dadaista`), digitando as entradas do programa manualmente, como também usando arquivos de entrada com listas maiores, para evitar L.E.R. Para isso, podemos chamar nosso programa no terminal com `python3 gerador_dadaista < entrada.txt`, onde o arquivo `entrada.txt` possui todos as listas e números pedidos pelo programa (um valor por linha), e a saída do programa aparecerá no próprio terminal; alternativamente, podemos usar `python3 gerador_dadaista < entrada.txt > saida.txt`, o que produzirá um arquivo de saída com a poesia gerada.

Leitura e armazenamento das palavras

O usuário deverá informar a quantidade de substantivos e em seguida digitar uma lista de substantivos precedidos pelos artigos respectivos, um substantivo por linha. Depois ele deve informar a quantidade de verbos e digitar os verbos com as preposições correspondentes (ou '-' se o verbo não pede preposição). Um exemplo de entrada possível é:

```
Quantos substantivos você deseja utilizar?
3
Digite um substantivo (com artigo) por linha:
o guarda-chuva
a cebola
a interrogação

Quantos verbos você deseja utilizar?
4
Digite um verbo (com preposição) por linha:
gosta de
engana -
viaja com
reza por
```

Construção das frases

A construção das frases envolve os seguintes aspectos, não necessariamente nesta ordem:

- a escolha das palavras;
- a declinação das preposições;
- a escolha da ordem dos termos na frase; e
- o tratamento das rimas.

A escolha das palavras será feita *obrigatoriamente* de maneira aleatória e evitando repetições. Já utilizamos a função *randint* da biblioteca *random* no primeiro EP:

```
from random import randint
randint(a,b) # gera aleatoriamente um número inteiro entre a e b
```

Para saber se uma palavra já foi usada ou não, você pode manter um vetor de indicadores booleanos para cada classe de palavras (substantivos e verbos). Cada sorteio pode então ser repetido um certo número de vezes até que se encontre uma palavra que ainda não foi usada. É importante observar que a não-repetição não é uma condição obrigatória, do contrário não seria possível gerar poemas longos a partir de um conjunto pequeno de palavras. Considere que cada sorteio pode ser repetido até um máximo de `M=númeroDeSubstantivos+númeroDeVerbos` vezes; se todas as palavras sorteadas já tiverem sido usadas, fica-se com a última palavra sorteada.

As frases dos poemas também podem começar com conjunções, tais como “como”, “e”, “enquanto”, “mesmo quando”, “porque”, “quando”, “se”, “toda vez que”, e outras mais que você quiser incluir. Estas conjunções deverão ser sorteadas aleatoriamente, e utilizadas pelo programa em um terço das frases do poema, sempre na posição inicial da frase.

Quando a frase possui um verbo transitivo indireto, devemos declinar a preposição correspondente de acordo com o artigo do objeto, através das regras habituais: a+a=à, a+o=ao, de+o=do, de+a=da, em+a=na, em+o=no, por+a=pela, por+o=pelo, enquanto outras preposições mantêm-se separadas dos artigos (com a, com o, para a, para o).

A escolha da ordem dos elementos principais da frase (com exceção da conjunção, quando existir) deve ser feita aleatoriamente, com probabilidade igual para os 3 casos seguintes (você pode considerar outras inversões, se quiser):

sujeito + verbo + objeto

sujeito + objeto + verbo

verbo + objeto + sujeito

A rima na realidade é um conceito complicado, mas para descomplicar nossa vida vamos considerar que duas palavras rimam quando elas terminam com as mesmas duas letras (fácil, né?). Se o usuário pedir poemas com rimas, utilizaremos o esquema de rimas AABBCCDD... ou seja, ao final das frases pares a palavra deve rimar com a última palavra da frase imediatamente anterior. Em outras palavras, nas frases ímpares não nos preocuparemos com rima, mas guardaremos a última palavra para gerar a frase seguinte; nas frases pares, após definir a ordem dos termos, vamos procurar uma palavra da classe correspondente ao terceiro termo (pode ser substantivo ou verbo) que rime com a palavra que guardamos da frase anterior.

Assim como a não-repetição, a obtenção de uma rima também será tratada com flexibilidade (afinal pode acontecer de não existir rima em alguns casos): você poderá sortear até `M` palavras tentando satisfazer as condições de não-repetição e rima. Se após M tentativas não for obtida nenhuma palavra nova com a rima certa, fica-se com a última palavra sorteada.

Formatação da saída

Para deixar os poemas visualmente mais bonitos, cuide dos seguintes detalhes:

- as frases ímpares devem começar com letra maiúscula;
- as frases pares devem terminar com ponto final;
- a última frase do poema também deve terminar com ponto final, mesmo se for uma frase ímpar;
- pule uma linha no final de cada frase, e deixe uma linha em branco a cada 4 frases.

Arquivos de Exemplo

No PACA você encontrará um programa executável com uma implementação deste EP. Use-o como referência para a sua implementação. Há também arquivos com substantivos e verbos que você pode usar como entrada, para evitar ficar digitando sempre as mesmas palavras.

Você pode criar e publicar no PACA arquivos com outros substantivos e verbos, por exemplo de temas particulares (futebol, política, astrologia, física quântica, crítica literária, gastronomia,...).

Last but not least...

Lembre-se sempre de:

- Ler as Instruções para Entrega de EPs no PACA;
- Não postar códigos no Fórum Geral, nem os compartilhar com colegas;
- Não deixar para submeter o EP para a última hora: o PACA fecha automaticamente o sistema de submissões no prazo definido. Você sempre pode submeter versões preliminares e depois substituí-las por versões melhores.
- Divertir-se programando!

Bom Trabalho!