Probabilidade das palavras geradas

Transcrição

Neste passo, calcularemos a **probabilidade** dos resultados gerados.

Pra construirmos a função `probabilidade()` em uma nova célula, primeiro precisaremos fazer este cálculo com uma palavra; a **frequência** do termo "lógica" será a quantidade de vezes que este parece no corpus em relação ao total de palavras. Para calcular, o `nltk` possui a função `.FreqDist()`.

Para isso, criaremos uma variável chamada `frequência` sendo igual ao método que calcula a **Distribuição de Frequência** dos vocábulos, e passaremos a `lista_normalizada` como parâmetro.

Na linha seguinte, *plotaremos* os dez primeiros itens escrevendo `frequencia` com `.most_common()` recebendo `10`.

```
frequencia = nltk.FreqDist(lista_normalizada)
frequencia.most_common(10)
```

COPIAR CÓDIGO

Executando o código, criaremos uma lista com as dez palavras mais frequentes no corpus. Desta forma, acessaremos a frequência de cada uma utilizando a anotação de dicionário.

Na célula seguinte, escreveremos `frequencia[]` e passaremos a palavra que queremos dentro dos colchetes, para então descobrirmos a frequência. Neste caso, passaremos o termo `"lógica"`.

```
frequencia["lógica"]
```

COPIAR CÓDIGO

Com isso, receberemos o valor de `87` vezes em que este vocábulo aparece no corpus.

Em seguida, construiremos a função de **`probabilidade()`** com `def` na próxima célula, e passaremos a `palavra_gerada`. Apenas relembrando, a função `max()` calculou a probabilidade chamando `probabilidade` e passando a palavra gerada pelo `gerador_palavras()`.

O `return` desta função será a `frequencia[]` da `palavra_gerada` dividido pelo `total_palavras`.

Em uma nova célula anterior a esta, criaremos a variável `total_palavras` sendo igual ao `len()` da `lista_normalizada`, da mesma forma como fizemos anteriormente para descobrirmos o valor de `17654` tipos diferentes de palavras.

```
total_palavras = len(lista_normalizada)
```

COPIAR CÓDIGO

```
def probabilidade(palavra_gerada):
    return frequencia[palavra_gerada]/total_palavras
```

COPIAR CÓDIGO

Estamos utilizando uma **variável global** porque a `probabilidade()` será chamada constantemente para retornar apenas os valores.

Para melhorarmos a visualização do código, recortaremos a linha de `total_palavras` e colaremos na mesma célula onde a `frequencia` foi definida anteriormente.

```
frequencia = nltk.FreqDist(lista_normalizada)
total_palavras = len(lista_normalizada)
frequencia.most_common(10)
```

COPIAR CÓDIGO

Também apagaremos a célula onde fizemos um teste com o termo `"lógica"` em `frequencia[]`.

Na função `probabilidade()`, calcularemos a probabilidade da palavra `“lógica”` no corpus, pois é a correta no caso do equívoco em "lgica".

```
def probabilidade(palavra_gerada):
    return frequencia[palavra_gerada]/total_palavras

probabilidade("lógica")
```

COPIAR CÓDIGO

Como retorno, teremos o número `0.00022` aproximadamente. Este valor pequeno se dá por conta do grande volume total de palavras, onde "lógica" aparece 87 vezes.

Na célula seguinte, calcularemos a probabilidade de uma outra palavra, como “lagica” por exemplo, a qual foi gerada como candidata a correção pelo algoritmo de inserção.

```
probabilidade("lagica")
```

COPIAR CÓDIGO

Sua probabilidade será `0`, afinal não há ocorrências deste termo no corpus. Ainda, a frequência da palavra “logica” sem acento é um número muito menor do que em relação à “lógica” com acento.

Isso acontece porque a autora ou autor do texto eventualmente pode esquecer de adicionar a acentuação correta em uma palavra. Porém isso também é um erro de digitação, e aparecem com bem menos frequência do que a palavra correta.

Portanto, nosso algoritmo será capaz de corrigir termos como "lgica" para “lógica”. Para testarmos, chamaremos a função `corretor()` com a `palavra_exemplo`.

```
corretor(palavra_exemplo)
```

COPIAR CÓDIGO

A saída do corretor será justamente a palavra `'lógica'` mais frequente, a qual é a correção para quando digitamos erroneamente “lgica”. Portanto, conseguimos completar o *pipeline* do corretor ortográfico para este primeiro exemplo.

Porém, só estamos corrigindo o tipo de erro onde uma letra não foi digitada, o termo “lgica”, e é interessante tratarmos outros casos.

Se por acaso digitarmos um caractere a mais, como "lógicaa" ou "lógicca" por exemplo, também deveremos ter **outras operações** que realizam a correção, além da inserção que já vimos.