
# Tokenização, Stemming, Lematização e Expressões Regulares em PLN

**INTEGRANTES:**
FELIPE RODRIGUES SANTANA
NICOLAS VIEIRA DOS SANTOS
VITORIA MARIA MENESES MOTA TEIXEIRA

**DISCIPLINA:** PROCESSAMENTO DE LINGUAGEM NATURAL

**REPOSITÓRIO:** https://github.com/vitoriameneses/PLN_2025_vitoria_nicolas_felipe.git

# What does Keras Tokenizer method exactly do?

**Etapa 1: Seleção da Pergunta no Stack Overflow**

**Critérios aplicados:**

- Pergunta deve envolver obrigatoriamente Tokenização e ou Stemming e ou Lematização e ou Expressões Regulares
- Pergunta com pelo menos 20 votos
- Deve ter pelo menos uma resposta aceita

**Passos executados:**

1. Busca no Stack Overflow com a tag **`nlp`** e filtros relacionados a tokenização.
2. Ordenação por “Highest score”.
3. Seleção da pergunta que satisfaz todos os critérios.

**Pergunta escolhida:**

- “What does Keras Tokenizer method exactly do?”
    - - 119 votos
    - - Possui resposta aceita com pontuação 188

**Etapa 2: Aplicando o Keras Tokenizer na prática**

A dúvida do autor da pergunta é sobre como funciona o método Tokenizer() do Keras, qual a diferença (caso exista) entre o fit_on_texts e texts_to_sequences. Na resposta aceita do StackOverflow, o usuário fornece uma breve explicação e também demonstra com o exemplo:

```
from tensorflow.keras.preprocessing.text import Tokenizer

texts = ['The cat sat on the mat.', 'The dog ate my homework.']

tokenizer = Tokenizer()
tokenizer.fit_on_texts(texts)

print("Vocabulário (word_index):")
```

```
print(tokenizer.word_index)

print("\nSequências:")
print(tokenizer.texts_to_sequences(texts))
```

A solução se divide nos seguintes passos:

1. Criar os textos que serão utilizados na tokenização, no exemplo acima foram escolhidas as frases: “The cat sat on the mat.” e “The dog ate my homework”.
2. Criar o Tokenizer e aplicar o fit_on_texts()
    a. O Tokenizer varre as palavras dos textos fornecidos.
    b. Cria um vocabulário com base na frequência de palavras.
    c. Atribui um índice para cada palavra, onde as mais frequentes possuem os menores índices.
    d. Por fim, a estrutura tokenizer.word_index armazena esse mapeamento.
3. Converter os textos em números. Cada palavra nas frases originais é substituída pelo número correspondente no word_index.

O dicionário gerado foi:

```
texts = {'the': 1, 'cat': 2, 'sat': 3, 'on': 4, 'mat': 5,
 'dog': 6, 'ate': 7, 'my': 8, 'homework': 9}
```

Podemos visualizar como o Tokenizer:

- Converte todas as palavras para minúsculas;
- Remove pontuação;
- Organiza o vocabulário por ordem de frequência;
- Converte frases em sequências numéricas prontas para redes neurais.

Após esse passo a passo, as frases foram convertidas em sequências de inteiros, prontas para serem usadas em modelos de deep learning.

Com outro exemplo do Tokenizer abaixo:

```
from tensorflow.keras.preprocessing.text import Tokenizer

texts = ['The quick brown fox jumps over the lazy dog.', 'Never jump
over the lazy dog quickly.', 'The dog is not just lazy, it is clever and
loyal.']

tokenizer = Tokenizer()
tokenizer.fit_on_texts(texts)

print("Vocabulário (word_index):")
print(tokenizer.word_index)
```

```
print("\nSequências:")
print(tokenizer.texts_to_sequences(texts))
```

O dicionário gerado foi:

```
texts = {'the': 1, 'dog': 2, 'lazy': 3, 'over': 4, 'is': 5,
 'quick': 6, 'brown': 7, 'fox': 8, 'jumps': 9, 'never': 10,
 'jump': 11, 'quickly': 12, 'not': 13, 'just': 14, 'it': 15,
 'clever': 16, 'and': 17, 'loyal': 18
}
```

Neste segundo exemplo, podemos também visualizar como o Tokenizer lida com repetições, já que nos textos selecionados para treiná-lo possuem palavras repetidas e também com diferenças de somente uma letra, o que pode fazer com que  seus índices variem, pois ocorrem empates na frequência.

Uma outra situação que pode ocorrer com o Tokenizer, é ao passar um novo texto para o texts_to_sequence, que contenha palavras fora do vocabulário do fit_on_texts.

```
from tensorflow.keras.preprocessing.text import Tokenizer

texts = ['The quick brown fox jumps over the lazy dog.', 'Never jump
over the lazy dog quickly.', 'The dog is not just lazy, it is clever and
loyal.']

tokenizer = Tokenizer()
tokenizer.fit_on_texts(texts)

print("Vocabulário (word_index):")
print(tokenizer.word_index)

print("\nSequências:")
print(tokenizer.texts_to_sequences(['The cat is not just lazy, it is
clever and loyal.']))
```

O texts utilizado foi o mesmo do exemplo anterior, dessa forma , o vocabulário gerado pelo fit_on_text se manteve igual, mas ao passar para texts_to_sequences uma nova expressão para testar o treinamento, ele acaba “pulando” a palavra que não é reconhecida (nesse caso, “cat”).

```
Sequências:
[[1, 5, 13, 14, 2, 15, 5, 16, 17, 18]]
```