



UNIVERSIDADE  
FEDERAL DO CEARÁ

# NLP com FastText

---

Mário Patrício

# Sobre o FastText

- Biblioteca para aprendizagem eficiente de classificação e representação de texto.
- É grátis e open-source.
- Funciona em hardware genérico padrão.
- Os modelos podem ser reduzidos em tamanho para caberem até mesmo em dispositivos móveis.

The logo for FastText, featuring the word "fast" in a red, italicized sans-serif font, followed by the word "Text" in a blue, bold sans-serif font.

Ele tem modelos pré-treinados em 157 línguas, **incluindo português**, baseados no Common Crawl!



# Aplicações do FastText

- Geração de embeddings.
- Busca por similaridade de palavras e frases.
- Treinamento de modelos textuais.
- Quantização e outras opções para redução de modelos.

The logo for FastText, featuring the word "fast" in a red, italicized sans-serif font, followed by the word "Text" in a blue, bold sans-serif font.

# Como utilizar - Instalações

```
pip install fasttext
```

```
!wget https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.pt.300.bin.gz
```

```
!gunzip cc.pt.300.bin.gz
```



# Como utilizar - utilidades Básicas

```
import fasttext
import numpy as np

model = fasttext.load_model("cc.pt.300.bin")

# Obter vetor de uma palavra
vec = model.get_word_vector("cachorro")
print(vec.shape)  # (300,)
```

# Como utilizar - utilidades Básicas

```
▶ def cosine(u, v):  
    return np.dot(u, v) / (np.linalg.norm(u) * np.linalg.norm(v))  
  
print(cosine(model.get_word_vector("homem"), model.get_word_vector("mulher")))
```

↔ 0.6712552



# Como utilizar - utilidades Básicas

```
# similaridade de frases

def sentence_vector(sentence):
    words = sentence.lower().split()
    vecs = [model.get_word_vector(w) for w in words if w in model.words]
    return np.mean(vecs, axis=0) if vecs else np.zeros((300,))

frase1 = "O carro é rápido"
frase2 = "O automóvel é veloz"

v1 = sentence_vector(frase1)
v2 = sentence_vector(frase2)

print("Similaridade:", cosine(v1, v2))
```

Similaridade: 0.9724057





# Como utilizar - utilidades Básicas

```
# Ver palavras mais similares
similares = model.get_nearest_neighbors("cachorro")
print(similares)
```

```
[(0.8647047877311707, 'cão'), (0.8203170299530029, 'cachorrinho'), (0.8168483376502991, 'cãozinho'),
```





UNIVERSIDADE  
FEDERAL DO CEARÁ

# NLP com FastText

---

Mário Patrício