
# Técnicas de PLN aplicadas a classificação de textos

# 1 - A representação Bag of Words e o problema da tokenização por espaços

# O problema da tokenização por espaços

- As pessoas podem pontuar de forma diferente.
- Frase 1:
  - Eu tive nota dez, pois estudei muito.
- A tokenização por espaços gera:
  - [“Eu”, “tive”, ”nota”, “dez,” ,”pois”, “estudei”, “muito.”]

- Frase 2:
  - Eu tive nota dez pois, estudei muito.
- A tokenização por espaços gera:
  - [“Eu”, “tive”, ”nota”, “dez”, ”pois,”, “estudei”, “muito”]

# O problema da tokenização por espaços

- A representação bag of words usa todos os tokens de todas as frases!
- Comparando as frases com bag of words o resultado é diferente quando deveria ser igual.
- Frase 1: Eu tive nota dez, pois estudei muito.

| Eu | tive | nota | dez, | pois | estudei | muito. | dez | pois, | muito |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

- Frase 2: Eu tive nota dez pois, estudei muito.

| Eu | tive | nota | dez, | pois | estudei | muito. | dez | pois, | muito |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |

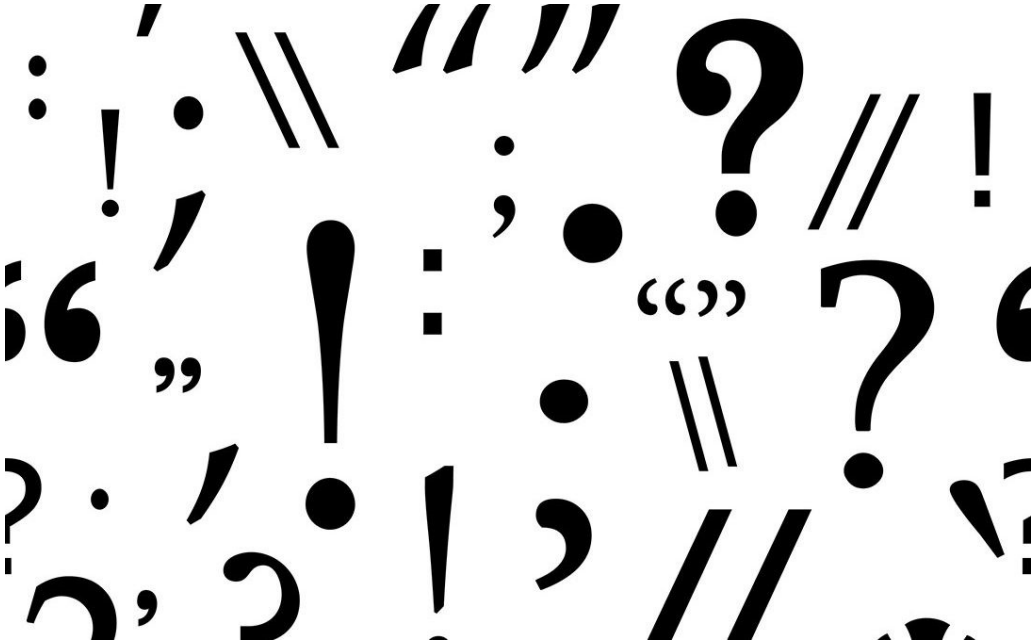
# Opções do pacote tokenize do NLTK

- from nltk.tokenize import **WhitespaceTokenizer**
- Tokeniza uma string pelos espaços em branco (espaço, tabulação, nova linha).

- from nltk.tokenize import **WordPunctTokenizer**
- Tokeniza uma string pelos espaços em branco (espaço, tabulação, nova linha) e caracteres não alfabéticos (“,”, “!”, “.”, “$” etc.)
  - Frase: O hambúrguer me custou R$ 30,00.
  - Tokens: [“O”, “hambúrguer”, “me”, “custou”, “R”, “$”, “30”, “,”, “00”, “.”]

# Pontuação como *stop words*

- As pontuações nos textos tem baixa relevância para a análise de sentimento.
- No nosso contexto as pontuações são stop words (“palavras” irrelevantes)
- punctuation é uma string com um conjunto de pontuações que vamos usar de referência para remover dos textos.



Unyleya
EDUCACIONAL

# 2 - O problema das variações de acentuação e de maiúsculas e minúsculas

# E a variação da acentuação?

- Acertando ou errando, é comum que as pessoas acentuem de forma diferente.
  - Frase 1:
    - Meu tênis é muito bonito.
  - Frase 2:
    - Meu tenis e muito bonito.
- O significado de ambas as frases é o mesmo, mas a aprendizagem automática pode não compreender isso.

# Representação de caracteres

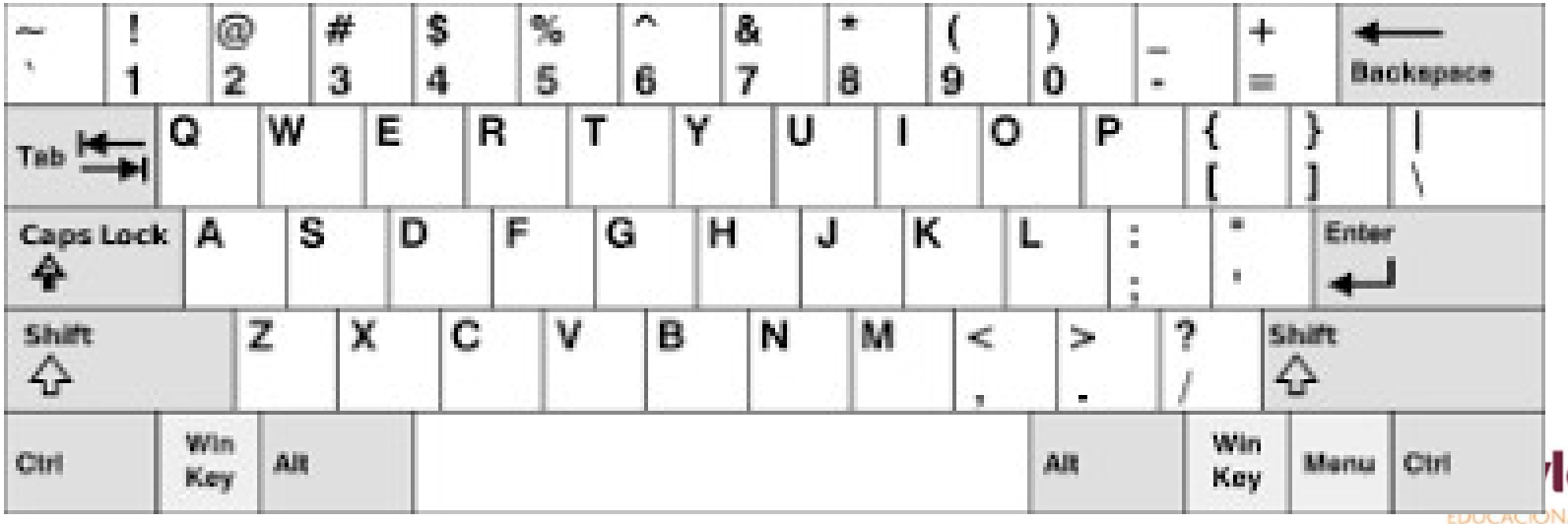
- Um caractere é o menor componente possível de um texto.
  - ‘A’, ‘B’, ‘C’, ‘È’ e ‘θ’ são exemplos.
- Os caracteres variam de acordo com o idioma ou contexto que você está falando.
- Unicode é um padrão usado no mundo todo para manipular texto de qualquer sistema de escrita.
- Em 2016 o Unicode tinha suporte para mais de 120.000 caracteres.

| | | | | |
|---|---|---|---|---|
| ¸ 00B8 | È 00C8 | Ø 00D8 | è 00E8 | ø 00F8 |
| ¹ 00B9 | É 00C9 | Ù 00D9 | é 00E9 | ù 00F9 |
| º 00BA | Ê 00CA | Ú 00DA | ê 00EA | ú 00FA |
| » 00BB | Ë 00CB | Û 00DB | ë 00EB | û 00FB |
| ¼ 00BC | Ì 00CC | Ü 00DC | ì 00EC | ü 00FC |
| ½ 00BD | Í 00CD | Ý 00DD | í 00ED | ý 00FD |
| ¾ 00BE | Î 00CE | Þ 00DE | î 00EE | þ 00FE |
| ¿ 00BF | Ï 00CF | ß 00DF | ï 00EF | ÿ 00FF |

Unyleya EDUCACIONAL

# Conversão de representação

- Muitos caracteres do Unicode são incompatíveis com aplicações especificas. Precisamos **normalizar** para os caracteres da aplicação.
- A biblioteca **Unidecode c**onverte dados Unicode para caracteres compatíveis com o que pode ser feito com um teclado no idioma inglês.
- Inglês não usa acentos, então removeremos acentos!



# Sensibilidade a maiúsculas e minúsculas?

- Sabemos que as frases abaixo vão ser tratadas como iguais quanto independente da acentuação:
  - Esse filme é horrível!
  - esse filme é horrível!
  - ESSE FILME É HORRÍVEL!
- Mas nossa classificação automática ainda está *case-sensitive!*
  - Esse != esse
  - filme != Filme
  - horrível != HORRÍVEL

# Sensibilidade a maiúsculas e minúsculas?

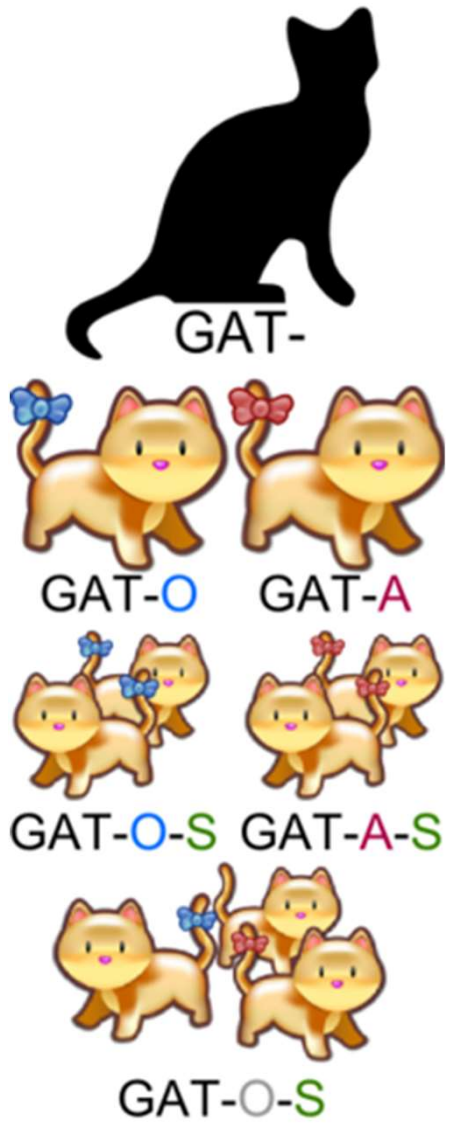
- A função lower() converte maiúsculas em minúsculas
- frase = "O Delorean é do Dr Brown"
- print(frase.lower())
  - resultado: `o delorean é do dr brown`

- Essa é a **normalização** em minúsculas, também chamada de *case folding*.

# 3 - Considerando as diversas flexões e derivações das palavras

# Flexões e derivações de palavras?

- Flexão é a modificação de uma palavra para expressar diferentes categorias gramaticais, como modo, tempo, voz, aspecto, pessoa, número, gênero e caso.
- As flexões criam pequenas diferenças no significado das palavras, criando novas palavras
  - vocabulários enormes.
- As pessoas erram as flexões com frequência.
  - confundem humanos e máquinas.



# Stemização!

- Processo de reduzir palavras flexionadas (ou derivadas) ao seu tronco (*stem*), base ou raiz.
- Nossa máquina verá as palavras ótimo, ótima, ótimos, ótimas, como a mesma palavra:
  - **otim**
- Nossa máquina verá as palavras péssimo, péssima, péssimos e péssimas como a mesma palavra:
  - **pessim**

# 4 - TF-IDF

# A frequência das palavras importa?

- O número de vezes que uma palavra ocorre em um determinado documento é chamado de frequência de termo, comumente abreviado como TF.
- Filme é uma palavra muito comum em textos sobre filmes, então tem alta frequência!
- Que palavra abaixo é mais importante para a análise de sentimentos?
  - Assisti um filme otimo.
  - Assisti um filme pessimo.

# A frequência das palavras importa!

- Assisti um filme otimo.

| Assisti | um | filme | ótimo | péssimo |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 |

- Assisti um filme pessimo.

| Assisti | um | filme | ótimo | péssimo |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 1 |

- As palavras filme , pessimo e otimo nao tem a mesma importancia mas tem o mesmo valor de frequencia em cada frase.

# Ponderando a frequência

- TF-IDF é uma ponderação das frequências para indicar as palavras mais informativas de um contexto.
- IDF pode ser traduzido como o inverso da frequência nos documentos.
  - Diz quantos documentos tem a palavra.
- O TF-IDF é a divisão do TF pelo IDF.

# Ponderando a frequência

- O TF-IDF padrão do conjunto de frases [“Assisti um filme otimo. , Assisti um filme pessimo]

| | Assisti | filme | péssimo | um | ótimo |
|---|---|---|---|---|---|
| Frase 1 | 0.448321 | 0.448321 | 0.000000 | 0.448321 | 0.630099 |
| Frase 2 | 0.448321 | 0.448321 | 0.630099 | 0.448321 | 0.000000 |

- Agora medimos quais palavras são mais informativas em cada frase!

# 5 - Ngrams

# A ordem das palavras importa pouco?

- As palavras isoladas não guardam toda a semântica de um texto.
  - Mas o filme não é ruim, é muito ruim.
  - Mas o filme não é ruim, ruim é muito.
- Ambas as frases tem as mesmas palavras, mas tem sentimentos diferentes.
- N-gram é uma sequência contendo n elementos.
- 1-gram, ou unigrama, tem 1 elemento, e usamos esses até agora.
- 2-gram, ou bigrama, tem 2 elementos.
- Tem algum bigrama que diferencie melhor nossas duas frases acima?

# Extraindo informação com ngram

- Bag of words com bigramas

| | Mas o | o filme | filme não | não é | é ruim | ruim é | é muito | muito ruim |
|---|---|---|---|---|---|---|---|---|
| Frase 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| Frase 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

- Com ngrams podemos extrair mais informação das frases
- Aumenta o vocabulário do aprendizado, cuidado!

Unyleya EDUCACIONAL

Obrigado!