Feature Selection

Seleção de Recursos

**Sumário**

[1. Introdução 3](#_Toc80544014)

[2. Estudo 3](#_Toc80544015)

[3. Estudar e analisar bigrama e trigrama 3](#_Toc80544016)

[4. Estudar a biblioteca Gensim com Python 3](#_Toc80544017)

[5. Testar exemplos 3](#_Toc80544018)

[6. Referências 4](#_Toc80544019)

# **Introdução**

**Aula dia ():** Aula

Um Sistema de recuperação de Informação (SRI) deve dentre outras funções ter uma base (corpus) com “todos” os documentos que se deseja trabalhar e deve possuir alguns rótulos específicos.

# **Estudos dos livvros**

**Livro:** Manning, Raghavan, Schutze - Introduction to information retrieval (2008), páginas 251 a 258 (275-)

**Livro:** Aggarwal - Machine Learning for Text (2018), páginas 117 a 122

***Seleção de recursos*** é o processo de seleção de um subconjunto dos termos que ocorrem em um conjunto de treinamento e usando apenas este subconjunto como recursos na classificação de texto.

Serve a dois propósitos principais.

1. Primeiro, torna o treinamento e a aplicação de um classificador mais eficientes ao diminuir o tamanho do vocabulário efetivo. Isso é de particular importância para classificadores que, ao contrário do NB (Naive Bayes), são caros de treinar.
2. Em segundo lugar, a seleção de recursos muitas vezes aumenta a classificação de acurácia eliminando recursos de ruído. Um recurso de ruído é aquele que, quando adicionado para a representação do documento, aumenta o erro de classificação em novos dados. Suponha um termo raro, digamos aracnocêntrico, não tem informações sobre uma classe, digamos China, mas todas as instâncias de aracnocêntrico aconteceu de ocorrer em documentos da China no conjunto de treinamento. Então, o método de aprendizagem pode produzir um classificador que atribui erroneamente documentos de teste contendo aracnocêntrico para China. Tal generalização incorreta de uma propriedade acidental do treinamento, isso é um **overfitting**.

# **Estudar e analisar bigrama e trigrama**

**xxx**

# **Estudar a biblioteca Gensim com Python**

**xxx**

# **Testar exemplos**

from gensim.models.phrases import Phrases

modelo\_fraseador = Phrases(frases, min\_count = min\_c, threshold = threshold,

scoring = scoring)

min\_c = 120, threshold = 0.6, scoring = 'npmi'

trabalhar com notícias de 2016 a 2021

[segunda-feira 15:53] Frederico Shu

Depois de treinar o modelo fraseador:

fraseador = Phraser(modelo\_fraseador) # exportar e fechar o modelo para economizar RAM

fraseador.save(arqfr) # gravar o modelo treinado

fraseador = Phraser.load(arqfr) # carregar fraseador

documento\_analisado = fraseador[documento]

A ideia então seria implementar o fraseador e estudar seleção de atributos

# **Bibliografia**

SRI

# **Referências**

<https://www.amazon.com.br/Introduction-Information-Retrieval-English-Christopher-ebook/dp/B00AHTN5JM>

Introdução à Recuperação da Informação

* <https://youtu.be/lJVw0-qbqEM>
* <https://www.youtube.com/watch?v=uwW_NTRpzC8>
* <https://www.youtube.com/watch?v=qWZbHr_GJgE>

ver COMO REALIZAR A SELEÇÃO DAS MELHORES VARIÁVEIS PARA SUA MÁQUINA PREDITIVA - FEATURE SELECTION <https://youtu.be/0bPBxDhvqlI>

ver <https://www.youtube.com/watch?v=4RGT2YRHERY&t=1258s>

vre <https://www.youtube.com/watch?v=Gzn6srbzU30>

ver <https://www.youtube.com/watch?v=0LyIGw8MToY&t=3270s>