REVISTA UIIPS

INSTITUTO POLITÉCNICO DE SANTARÉM

9 - 4 | 2021

Processamento de linguagem natural aplicado à inteligência artificial

Natural language processing applied to artificial intelligence

Procesamiento del lenguaje natural aplicado a la inteligencia artificial

André Moreira | Artur Marques | Cristiana Modesto | Nádia Nobre | Ricardo Tiago Silva | Ricardo Silva | Wilson Oliveira

Electronic version
URL: https://revistas.rcaap.pt/uiips/ ISSN: 2182-9608

Publisher
Revista UI_IPSantarém

Printed version
Date of publication: 31st December 2021 Number of pages: 10
ISSN: 2182-9608

Electronic reference

Moreira, A.; Marques, A.; Modesto, C.; Nobre, N.; Silva, R. T.; Silva. R. & Oliveira, W. (2021). *Processamento de linguagem natural aplicado à inteligência artificial.* Revista da UI_IPSantarém. *Edição Temática: Ciências Exatas e das Engenharias*. Número especial: Conferência Internacional Cooperação Internacional, multiculturalidade, trabalho colaborativo e ambientes mais inclusivos, sustentáveis e resilientes. 9(4), 81-90. https://revistas.rcaap.pt/uiips/

revistauiips@ipsantarem.pt
Complexo Andaluz,
Apartado 279
2001-904 Santarém

Revista da UI_IPSantarém – Unidade de Investigação do Instituto Politécnico de Santarém,
Vol. 9, N. º 4, 2021, pp. 81-90, Santarém
ISSN: 2182-9608
https://revistas.rcaap.pt/uiips/

PROCESSAMENTO DE LINGUAGEM NATURAL APLICADO À INTELIGÊNCIA ARTIFICIAL

Natural language processing applied to artificial intelligence

Procesamiento del lenguaje natural aplicado a la inteligencia artificial

André Valente Moreira
Instituto Politécnico de Santarém
Escola Superior de Gestão e Tecnologia Santarém, Portugal
170100212@esg.ipsantarem.pt | https://orcid.org/0000-0001-8804-9930

Artur Marques
Instituto Politécnico de Santarém
Escola Superior de Gestão e Tecnologia de Santarém, Portugal
artur.marques@esg.ipsantarem.pt | https://orcid.org/0000-0002-1625-0341 |
Ciência ID: 5114-3496-0D1F

Cristiana Manuela Nunes Modesto
Instituto Politécnico de Santarém,
Escola Superior de Gestão e Tecnologia de Santarém, Portugal
190100131@esg.ipsantarem.pt | https://orcid.org/0000-0003-3897-8613

Nádia Miriam Simões Nobre
Instituto Politécnico de Santarém,
Escola Superior de Gestão e Tecnologia de Santarém, Portugal
190100236@esg.ipsantarem.pt | https://orcid.org/0000-0003-2189-4632

Ricardo Tiago Gregório da Silva
Instituto Politécnico de Santarém,
Escola Superior de Gestão e Tecnologia de Santarém, Portugal
190100365@esg.ipsantarem.pt | https://orcid.org/0000-0003-1790-6952 |
Ciência ID: 8B12-89FA-F0B4

Ricardo Ferreira Bastos Silva

Instituto Politécnico de Santarém

Escola Superior de Gestão e Tecnologia de Santarém, Portugal

170100190@esg.ipsantarem.pt | https://orcid.org/0000-0002-8424-2968

Wilson Cristiano Oliveira

Instituto Politécnico de Santarém,

Escola Superior de Gestão e Tecnologia de Santarém, Portugal

180100286@esg.ipsantarem.pt | https://orcid.org/0000-0003-4327-0396

RESUMO

Este artigo tem a finalidade de apresentar o tópico denominado por “Processamento de linguagem natural aplicado à inteligência artificial”. Tem como objetivos divulgar os seguintes itens: Conceitos, nomeadamente o Teorema de Bayes, Jaccard Index, Matplotlib, NLP, e ferramentas como o NLTK; “Chi-Square”, “NumPy” - onde será abordada a definição de cada; “SpaCy” – onde serão explicados os conceitos inerentes a este pacote. Para tal recorreu-se à revisão bibliográfica para a pesquisa de artigos relacionados com o exemplo inerente ao processamento de linguagem natural com “Machine Learning”, e utilizou-se um exemplo de um algoritmo de estilometria realizado no âmbito da unidade curricular de Inteligência Artificial, lecionada pelo docente Artur Marques.

Palavras-chave: Inteligência Artificial, Estilometria, Machine Learning, Natural Language Processing.

ABSTRACT

This article aims to present the topic called "Natural language processing applied to artificial intelligence". Its objectives are to disseminate the following items: Concepts, namely the Bayes Theorem, Jaccard Index, Matplotlib, NLP, and tools such as NLTK; "Chi-Square", "NumPy" - where will be addressed the definition of each; "SpaCy" – where the concepts inherent to this package will be explained. For this purpose, the bibliographic review was used for the research of articles related to the example inherent to the processing of natural language with "Machine Learning", and an example of a Stylometry algorithm performed within the artificial intelligence curricular unit, taught by Professor Artur Marques, was used.

Keywords: Artificial Intelligence, Stylometry, Machine Learning, Natural Language Processing.

1 INTRODUÇÃO

No âmbito da unidade curricular de Inteligência Artificial, lecionada pelo docente Artur Marques, foi sugerido apresentar o tópico denominado por “Processamento de linguagem natural aplicado à inteligência artificial”.

Os objetivos deste artigo são divulgar os tópicos: conceitos, nomeadamente o Teorema de Bayes, Jaccard Index, Matplotlib e NPL, e ferramentas como NLTK; Chi-Square – a estatística será abordada com os seguintes itens, definição e fórmula; NumPy - a definição; SpaCy – Elucidar os

conceitos inerentes a este pacote; Algoritmo de estilometria- criação de um modelo de classificação de autor, que analisa pedaços de texto de um livro de autor desconhecido, e, baseado nos dados de treino, determina a qual autor esse texto pertence.

Para tal recorreu-se à revisão bibliográfica para a pesquisa de artigos relacionados com o pacote inerente ao processamento de linguagem natural com “Machine Learning”, e utilizou-se um exemplo de um algoritmo de estilometria realizado no âmbito da unidade curricular de Inteligência Artificial.

2 CONCEITOS E EXEMPLOS DA APLICAÇÃO DO PROCESSAMENTO DE LINGUAGEM NATURAL À ESTILOMETRIA

Nesta secção serão descritos conceitos e um exemplo da aplicação do processamento de linguagem natural à estilometria.

2.1 Conceitos e ferramentas

Aqui serão descritos os conceitos aplicados implicitamente no desenvolvimento do código de exemplo desta secção e as ferramentas utilizadas no mesmo.

2.1.1 Conceitos

Neste tópico apresentam-se os conceitos aplicados no algoritmo de exemplo.

2.1.1.1 Índice de Jaccard

O índice de Jaccard é uma estatística usada para medir a similaridade e diversidade entre conjuntos de amostras. Muito usado na ciência da computação, o coeficiente mede a similaridade entre conjuntos de amostras finitas e é definido como a cardinalidade da interseção sobre a cardinalidade da união dos conjuntos.

O índice de similaridade ou coeficiente de similaridade foi criado por Paul Jaccard originalmente e pode ser expressa na seguinte notação:

$$J(A, B) = |A \cap B| / |A \cup B|$$

A e B são quais queres conjuntos em que |A| e |B| têm os seus próprios elementos.

Já a distância de jaccard mede a diferença entre os conjuntos de amostras e pode ser obtido subtraindo ao coeficiente de jaccard, 1. Como pode ser visto na expressão escrita abaixo.

$$Dj(A,B) = 1 – J|A,B|$$

2.1.1.2 Teorema de Bayes

Para a análise estatística, existe duas maneiras de abordar os problemas, a inferência de frequência ou o método Bayesiano. Na estatística de frequência, são unicamente contabilizados parâmetros que apenas não se alteram, mesmo que a experiência fosse repetida inúmeras vezes. Na inferência Bayesiana, existe o cuidado de usar fatores de interesse que na estatística convencional não são observados e que podem impactar negativamente o grau de certeza da experiência.

Em síntese, a estatística Bayesiana utiliza o entendimento precedente, ao contrário da clássica, que apenas considera parâmetros isolados sem ter em conta a existência de outros fatores, que no quotidiano poderiam influenciar os resultados da experiência, quando repetida várias vezes.

Para as experiências, deve-se ter em conta que existem sempre dados que contêm um grau de vulnerabilidade ou de incerteza e assim, com base nesse aspeto onde o erro é compreendido dentro do próprio conceito de probabilidade da estatística Bayesiana, esta está justamente ligada a um

elevado grau de confiança do resultado obtido do conjunto de dados observados. Por isso mesmo, cada experiência é uma experiência isolado e qualquer outra que venha em seguida, o resultado pode ter uma conclusão distinta.

Em termos de cálculo, para qualquer probabilidade de um acontecimento A se dar sabendo que o acontecimento B ocorreu pela estatística clássica, temos que:

P (A|B) = P(A∩B) / P(B) sendo B maior que 0

Já na estatística Bayesiana, para qualquer probabilidade de um acontecimento A se dar sabendo que o acontecimento B ocorreu, temos que:

P(A|B) = P(B|A) * P(A) / P(B)

Ao contrário da inferência frequentista precisamos de analisar mais alguns dados como:

P(B|A): Probabilidade de B acontecer se A ocorreu

P(A): Probabilidade de A acontecer

P(B): Probabilidade de B acontecer

2.1.1.3 Chi-Square

A estatística Chi-Square por definição representa o resultado de um teste para comparar o quão fiel um modelo representa dados observados sendo calculado pela seguinte formula:

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Oi – Valores observados

Ei – Valores esperados pelo modelo

2.1.1.4 NLP

O NLP ou Natural Language Processing permite que uma máquina consiga interagir com seres humanos através de computação linguística e tecnologias de forma a construir uma inteligência que se expresse numa linguagem natural.

2.1.2 Ferramentas

Neste tópico apresentam-se as ferramentas aplicadas no algoritmo de exemplo mencionado no tópico anterior.

2.1.2.1 NumPy

O NumPy é uma ferramenta da linguagem Python que se baseia no objecto array Numeric, este tem como objetivo criar a base para um ambiente útil para a computação científica.

2.1.2.2 NLTK

NLTK ou natural Language Tool Kit, é ferramenta desenvolvida para construir programas em Python que pretendam trabalhar como dados de linguagem natural (humana).

2.1.2.3 Matplotlib

Como todos os dados que são tratados são números, o mais confortável seria poder representar esses mesmos dados num modo gráfico e é isso mesmo que nos permite o Matplotlib, que é biblioteca para visualização de dados e impressão dos mesmos graficamente.

2.2 Determinação do autor de um texto tendo por base o conceito de estilometria

De seguida, tem-se um exemplo da atribuição de classificação de autor, analisando um pedaço de texto de um livro de autor desconhecido e baseando em pedaços de texto ambos de autor conhecido e em medidas calculadas, consegue demonstrar a qual autor é mais provável o texto pertencer.

2.2.1 Primeira Etapa

Obtenção de texto para autores conhecidos e para o autor desconhecido:

2.2.1.1 Textos de autores conhecidos:

```
“ (Método que lê o ficheiro .txt)

def readTextFileToString(pStrFileName):

    fr = open(pStrFileName, 'rt') #fr for file reader ; 'rt' is the default

    strFileContents = fr.read()

    return strFileContents

#def end “

strHound = readTextFileToString("hound.txt") #str

strWarOfTheWorlds = readTextFileToString("war.txt")
```

Os textos de autores conhecidos estão ambos no formato String.

2.2.1.2 Texto de autor desconhecido

```
strLostWorld = readTextFileToString("lost.txt")
```

O texto de autor desconhecido está também no formato String.

A partir daqui, é possível a realização de operações como:

- A tokenização, isto é, organizar cada String dos autores conhecidos e do autor desconhecido em conjuntos de tokens (partes da String);
- Determinar qual o texto com o menor número de palavras, essencial às bibliotecas mencionadas no ponto 2.2.2;
- E determinar o índice de jaccard e o Chi-Square, que indicam proximidade e distância dos dados, respetivamente.

A partir dos dados obtidos, é possível assim concluir a quem pertence o texto desconhecido.

2.2.2 Segunda Etapa

Este modelo, sem as bibliotecas abaixo indicadas, apenas consegue ler ficheiros “.txt” e converter o conteúdo desse ficheiro para uma variável String. A biblioteca “nltk” provém do termo “Natural Language Tool Kit”, o “nltk.corpus” é uma subclasse, e “stopwords” é uma função que vive na subclasse da biblioteca. Resumidamente, esta função vai permitir retirar de certo texto, as palavras que não dão grande valor á frase, ou seja, sem as mesmas, é possível determinar na mesma a quem pertence o texto. É importada também a biblioteca “matplotlib”, e uma das suas classes “pyplot”, que permite mostrar os resultados os resultados graficamente, onde consequentemente conseguimos dispor os dados de forma mais organizada e confortável.

2.2.3 Terceira Etapa

O código abaixo, resumidamente, calcula o índice de jaccard, que é uma medida de proximidade:

```
def jaccardIndexBasedOnVocabularyByAuthor(pDictAuthorsToTheirListOfTokens, piLengthOfShortestText):
    dictAuthorsToJIndex = {}
    listForUnknownAuthorTruncated = pDictAuthorsToTheirListOfTokens['UNKNOWN'][:piLengthOfShortestText]
    setUniqueTokensForUnknownAuthor = set(listForUnknownAuthorTruncated)

    for strAuthor in pDictAuthorsToTheirListOfTokens.keys():
        bUnknown = strAuthor == "UNKNOWN"
        if (not bUnknown):
            listTokensForCurrentAuthorTruncated = pDictAuthorsToTheirListOfTokens[strAuthor][:piLengthOfShortestText]
            setUniqueTokensForCurrentAuthor = set(listTokensForCurrentAuthorTruncated)
            setIntersectionOfCurrentAuthorWithUnknownAuthor = \
                setUniqueTokensForCurrentAuthor.intersection(setUniqueTokensForUnknownAuthor)
            setReunionOfCurrentAuthorWithUnknownAuthor = \
                setUniqueTokensForCurrentAuthor.union(setUniqueTokensForUnknownAuthor)
            iCardinalityIntersection = len(setIntersectionOfCurrentAuthorWithUnknownAuthor)
            iCardinalityReunion = len(setReunionOfCurrentAuthorWithUnknownAuthor)
            fJaccardIndex = iCardinalityIntersection / iCardinalityReunion
            dictAuthorsToJIndex[strAuthor] = fJaccardIndex

            strFormat = "%s => Jaccard-index= %.3f" % (strAuthor, fJaccardIndex)
            print(strFormat)
        #if not the unknown author
    #for every author
    return dictAuthorsToJIndex
#def jaccardIndexBasedOnVocabularyByAuthor
```

Figura 1: Código que permite calcular o índice de jaccard

2.2.4 Quarta Etapa

O código que se segue calcula a medida de distância Chi-Square:

```
def chiSquareBasedOnVocabularyByAuthor(pDictAuthorsToTheirListOfTokens):
    dictAuthorsToChiSquare = {}
    for strAuthor in dictAuthorsToTheirTexts.keys():
        bUnknown = strAuthor=="UNKNOWN"
        if (not bUnknown):
            #the texts
            listTokensForCurrentAuthor = pDictAuthorsToTheirListOfTokens[strAuthor]
            listTokensForAuthorUnknown = pDictAuthorsToTheirListOfTokens["UNKNOWN"]
            listTokensForBothTextsCombined = listTokensForCurrentAuthor + listTokensForAuthorUnknown

            #the size of the texts
            iHowManyTokensForCurrentAuthor = len(listTokensForCurrentAuthor)
            iHowManyTokensInCombinedText = len(listTokensForBothTextsCombined)
            fCurrentAuthorProportionInCombinedText = iHowManyTokensForCurrentAuthor / iHowManyTokensInCombinedText

            #frequency distribution and most common 1000 words in combined text
            freqDistForCombinedTexts = nltk.FreqDist(listTokensForBothTextsCombined) # nltk.probability.FreqDist
            listMmostFrequentWordsInCombinedTexts = freqDistForCombinedTexts.most_common(1000)
            # #the 1000 most frequent words
            listOfTuplesWordCountForMostFrequentWordsInCombinedText = list(listMmostFrequentWordsInCombinedTexts)

            chiSquareForCurrentAuthor = 0
            for (strCommonWordFromCombinedText, iWordCountInCombinedText) \
                in listOfTuplesWordCountForMostFrequentWordsInCombinedText:
                iHowManyTimesTheWordAppearsInTheCurrentAuthorText = listTokensForCurrentAuthor.count(strCommonWordFromCombinedText) #this is the "observed value"

                # if the word appears this number of times in the author's text
                # and if we know the author's text is present in the combined corpus with fCurrentAuthorProportionInCombinedText
                # then we would expect to find fCurrentAuthorProportionInCombinedText * count(strWord) , in the combined text
                fExpectedTimesTheWordAppearsInTheCombinedText = fCurrentAuthorProportionInCombinedText * iWordCountInCombinedText
```

Figura 2: Primeira parte do código que permite calcular a medida de distância Chi-Square

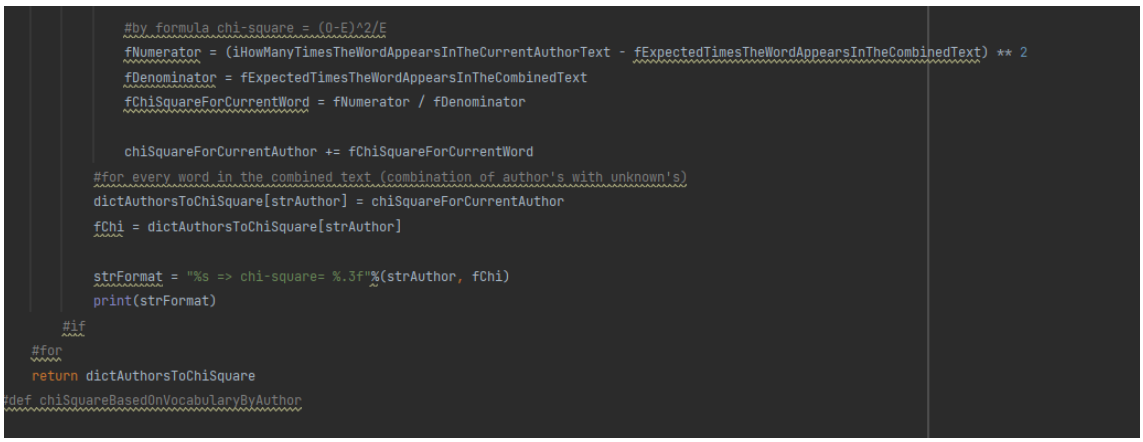

Figura 3: Segunda parte do código que permite calcular a medida de distância Chi-Square

3 CONCEITOS DA APLICAÇÃO DO PROCESSAMENTO DE LINGUAGEM NATURAL AO MACHINE LEARNING

No próximo ponto segue-se a apresentação dos conceitos do processamento de linguagem natural aplicado ao machine learning recorrendo ao package spaCy.

3.1 Conceitos inerentes ao processamento de linguagem natural aplicado à área de machine learning recorrendo ao package spaCy

Neste tópico, iremos abordar uma nova biblioteca que surgiu durante a pesquisa, denominada “spaCy”, que tem a particularidade de ter suporte para a língua portuguesa.

O processamento de linguagem natural, utiliza ferramentas da linguística para processar o texto da forma mais correta possível. Ao longo da leitura, foi possível a familiarização com o conceito de “Tokens” e Tokenização expostos na primeira secção, portanto para além da tokenização, irá agora ter foco a pós tokenização, que, no caso do Spacy faz com que o texto tokenizado entre numa “Trained Pipeline” resultando num objeto doc com as propriedades obtidas em cada componente da pipeline. A “Trained Pipeline” será traduzida como canal de treino.

O canal de treino é constituído pelos seguintes componentes:

Tokenizer- onde ocorre a tokenização de uma frase, resultando em “tokens” que são incluídos num objeto doc.

Tensorizer- onde ocorre a codificação da representação interna do “doc” como um “array” de “floats”, designado por tensor, uma vez que os modelos “spaCy” são do tipo redes neuronais.

Tagging- onde a cada “token” atribui-se a sua distribuição a nível sintático e morfológico (ex. nome, verbo), podendo surgir classificações tais como:

DET-determinante, NOUN-nome, ADJ-adjetivo, AUX- verbo auxiliar, ADP- adposições(traduzido do inglês “adposition”), por exemplo, preposições.

Parser- onde ocorre a análise de uma “String” de símbolos, para entender as dependências entre eles.

NER (Named entities recognition) - onde se faz o reconhecimento de nomes de entidades, por exemplo, pessoas, países, organizações. Denote-se que é possível determinar o tipo de entidade, através da propriedade “ents” do objeto “doc” para obter os “tokens” cujos nomes referem-se a entidades presentes na frase, e a cada um adicionar a tag “label_” para demonstrar o tipo de entidade.

Estes componentes tomam decisões baseados em modelos estatísticos que atribuem pesos aos exemplos recebidos (weight Value)

Weight Value – é o peso atribuído a determinada característica para ter em conta no resultado. No processo da máquina atribuir pesos: recebe-se os dados de treino (exemplos) e procura-se que pesos se pode atribuir às características para que estas coincidam com o output de cada um dos dados de treino.

De seguida, exemplifica-se o carregamento de um modelo de linguagem, no idioma pretendido, neste caso, português, sendo que este modelo será usado para processar o texto através do canal de treino:

Figura 4: Carregamento do modelo de linguagem em português

Alguns componentes do canal de treino do spaCy podem ser treinados pelo utilizador, como, por exemplo, o Tagger que faz a marcação gramatical do input recebido e o EntityRecognizer.

4 MÉTODO

Para a obtenção dos conceitos relacionados com cada algoritmo, recorreu-se à revisão bibliográfica, e ainda, a outros métodos de pesquisa para obtenção de informação.

Com a finalidade de obter algoritmos de exemplo para os conceitos abordados, recorreu-se a um exemplo na área da estilometria efetuado no âmbito da unidade curricular de Inteligência Artificial, e a um exemplo do package “spaCy” através da revisão bibliográfica.

5 RESULTADOS

Os resultados obtidos vão de encontro aos conceitos e aos algoritmos representados nas imagens e excertos de código, nas secções 2 e 3, onde é possível desta forma perceber o processamento de linguagem natural aplicado à área da estilometria, no caso da secção 2, e a aplicação à área de “Machine Learning” na secção 3.

6 DISCUSSÃO DE RESULTADOS

Os resultados obtidos permitem aprofundar a interpretação das áreas de estilometria e de “Machine Learning”, permitindo nesta última conhecer o package “spaCy”. Ambos tendo por base o conceito de processamento de linguagem natural.

7 CONCLUSÃO

Com este artigo, conclui-se que no âmbito da linguagem “Python”, o processamento de linguagem natural (PLN), que abrange a compreensão e a produção de linguagem natural, dispõe de vários módulos e ferramentas para a construção de algoritmos, que permitem experimentar e explorar técnicas como o “Machine Learning” e a Estilometria aplicados ao PLN.

8 REFERÊNCIAS

Connelly, L. (2019). Chi-square test. Medsurg Nursing, 28(2), 127–127. (Connelly, 2019)

Costa, L. da F. (2021). Further Generalizations of the Jaccard Index. ArXiv:2110.09619 [Cs]. http://arxiv.org/abs/2110.0961

GPE. (2019). Em Wikipedia. https://en.wikipedia.org/w/index.php?title=GPE&oldid=876300060 («GPE», 2019)

Hayes, A. (2021, setembro 20). Chi-Square (χ2) Statistic Definition. Investopedia. https://www.investopedia.com/terms/c/chi-square-statistic.asp

Jaccard index. (2021). Em Wikipedia. https://en.wikipedia.org/w/index.php?title=Jaccard_index&oldid=1040991564

Maynard, D., Bontcheva, K., & Cunningham, H. (2003). Towards a semantic extraction of named entities. (Maynard et al., 2003)

Nils, J. N. (2005). INTRODUCTION TO MACHINE LEARNING.

Oliphant, T. E. (2006). A guide to NumPy (Vol. 1). Trelgol Publishing USA.

Preface. (sem data). Em https://www.nltk.org/book/ch00.html

Portuguese · spaCy Models Documentation. (sem data). Portuguese. Obtido 3 de Dezembro de 2021, de https://spacy.io/models/pt (Portuguese · SpaCy Models Documentation, sem data)

Rao, C. (2002). Karl Pearson chi-square test the dawn of statistical inference. Em Goodness-of-fit tests and model validity (pp. 9–24). Springer.

Recursos da disciplina de Inteligência Artificial, lecionada pelo professor Artur Marques, no ano letivo 2021/2022

Rish, I. (sem data). An empirical study of the naive Bayes classifier. 6.

Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press. https://doi.org/10.1017/CBO9781107298019

Srinivasa-Desikan, B. (2018). Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras. Packt Publishing Ltd. (Srinivasa-Desikan, 2018)

Stenetorp, P., Pyysalo, S., Topic, G., Ohta, T., Ananiadou, S., & Tsujii, J. (sem data). brat: A Web-based Tool for NLP-Assisted Text Annotation. 6.

Straková, J., Straka, M., & Hajič, J. (2014). Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 13–18. https://doi.org/10.3115/v1/P14-5003

Thanaki, J. (2017). Python Natural Language Processing. Packt Publishing Ltd.

Timpani, V. D. (sem data). Uma Breve Introdução à Estatística Bayesiana Aplicada ao Melhoramento Genético Animal. 59.