Inteligência Artificial

Thiago Henrique Leite da Silva, RA: 139920

AULA11: Exercício prático PLN

1) Considere o seguinte arquivo textual:

https://www.kaggle.com/crawford/20-newsgroups

a) Escolha 2 tópicos dentre os 20 disponíveis no dataset e faça download dos mesmos.

Os dois tópicos escolhidos foram:

- soc.religion.christian.txt
- sci.crypt.txt

b) Gere um Bag of words: Leia o texto e armazene cada palavra em uma posição em um vetor. Faça a contagem da frequência de cada palavra gerando uma matriz termo-frequência. Na última coluna armazene o rótulo do texto (‘1’ = textos do tópico 1 e ‘2’ = textos do tópico 2)

c) Remova os stop words (palavras irrelevantes):

https://gist.github.com/sebleier/554280

Os passos B e C foram feitos em conjunto, o que fiz foi primeiramente dividir os textos e palavras, após a divisão, removi todos os caracteres que não fossem letras, posteriormente fiz a contagem de palavras por texto em um Hash do tipo (chave -> valor) sendo (palavra -> frequência). Após a contagem, juntei todas as palavras dos dois textos, não permitindo repetições, e coloquei em um CSV para visualizarmos a saída, portanto, o Bag Of Words está na planilha anexada.

Na hora de fazer a limpeza dos caracteres, já removia as stop words que estavam em um dataset que utilizei do Kaggle.

d) Escolha dois algoritmos de classificação vistos anteriormente (knn, naive bayes, arv. Decisão, svm, etc) e classifique os textos (separar 70% para treino e 30% para teste). Anexar a saída e % de acerto de cada algoritmo.

Os algoritmos que escolhi usar foi o KNN e o Naive Bayes, porém só tive tempo de implementar um deles, o Naive Bayes. Utilizei o mesmo esquema visto em aula, calculei as palavras de cada um dos textos, essa definição foi feita com base na frequência, ou seja, se a palavra A aparece mais vezes no texto 01 do que no texto 02, esta palavra A é considerada do texto 01, ao fazer as comparações, o algoritmo dará uma certa vantagem para o texto 01 ao se deparar com esta palavra.

Além disso, também calculo a quantidade de palavras do nosso vocabulário, que foram mais de 25mil.

Por fim, treinei o algoritmo com 13 frases de entrada e 6 frases de teste para que ele indicasse o texto a qual ela pertence. O resultado foi muito bom, o algoritmo passou em todos os testes.

Implementei na mão o código, por ser um dos primeiros contatos com a criação de classes em Python, acredito que esteja ruim e com ausência de algumas boas práticas da linguagem, mas foi uma experiência bem bacana, consegui aprender bastante.

Peço a gentileza de analisar com calma o código para entender todo o funcionamento professora, infelizmente não consegui fazer a documentação do mesmo por conta do tempo.

Segue o link no Kaggle com a implementação: https://www.kaggle.com/thiagohenriqueleite/pln-with-naivebayes-bagofwords

```
NaiveBayes(bag_of_words, bag_of_words_phrases, tests, tests_results).perform()
```

```
Processamento de Linguagem Natural

Algoritmo Naive Bayes

Texto 01: soc.religion.christian
Texto 02: sci.crypt

Foram utilizadas 13 frases para treino e 6 frases para teste

Tamanho do vocabulário utilizado: 26373 palavras.

Palavras com mais ocorrências no Texto 01: 13254
Palavras com mais ocorrências no Texto 02: 13120

Probabilidade de uma frase ser do Texto 01: 0.54 %
Probabilidade de uma frase ser do Texto 02: 0.46 %


Execução dos testes:

Frase: 'the bible teaches that no one is good enough in himself to go to heaven':
Classificação: Texto 01 ✓

Frase: 'i am looking for references to algorithms which can be used for password encryption':
Classificação: Texto 02 ✓

Frase: 'i study religion and read the bible':
Classificação: Texto 01 ✓

Frase: 'what i meant was that as long as the only advantage of the cryptanalyst is a faster computer':
Classificação: Texto 02 ✓

Frase: 'then it will be known to the ends of the earth that god rules':
Classificação: Texto 01 ✓

Frase: 'the fundamental strength of the des and rsa are not nearly so important as what we know about their strength':
Classificação: Texto 02 ✓


Acurácia do algoritmo: 6/6 acertos.
100.0%
```