

Relevância de Textos com Lógica Fuzzy

Eric Calasans de Barros José Genilson da Silva Filho

Universidade Federal do Rio Grande do Norte

4 de dezembro de 2017

Ferramentas Utilizadas

- **Python:**

- `nltk` - para análise e processamento do texto

- `skfuzzy` - para calcular relevância do texto através da lógica fuzzy

- **Textos para validação**

Metodologia

Relevância de
Textos com
Lógica Fuzzy

*Eric Calasans
de Barros,
José Genilson
da Silva Filho*

Capa

Introdução

Metodologia

Resultados

```
1 stemmer = RSLPStemmer() #extracao dos radicais das
   palavras
2 palavras = ['jogador', 'futebol'] # dicionario
3 arquivo = open('medicina.txt', 'r') #abrindo o
   arquivo
4 texto = arquivo.read() \# texto a ser comparado
5 words = (nltk.word_tokenize(texto)) #texto
   tokenrizsdo
6 stops = set(stopwords.words("portuguese")) #conjunto
   de palavras sem importancia para a
   classificacao do texto
7 word_features = ([w for w in words if not w in stops
   ]) #stop words removioda do texto
```

Listing 1: Preparação do Texto

Cálculo dos Matches

■ Match Total

$$matchTotal_i = \frac{nOcorrCompleta_i}{\sum_{i=1}^n nOcorrCompleta_i}$$

■ Match Radical

$$matchRadical_i = \frac{nOcorrRad_i}{\sum_{i=1}^n nOcorrRad_i}$$

■ No Match

$$noMatch_i = \frac{ausRad_i}{\sum_{i=1}^n ausRad_i}$$

Metodologia

Relevância de
Textos com
Lógica Fuzzy

*Eric Calasans
de Barros,
José Genilson
da Silva Filho*

Capa

Introdução

Metodologia

Resultados

```
1 #Procurando match total
2 def matchWord(texto , dicionario):
3     bag_words ={} # Inicia o dicionario que vai
4     guardar os matchs do dicionario no texto
5     for i in dicionario:
6         bag_words[i] = 0
7
8 # Procura os matchs entre o dicionario e o texto;
9 # incrementa +1 quando encontra na chave do
10 # dicionario que
11 # se refere a palavra que esta sendo analisada
12 for palavraTexto in texto:
13     for palavraDicionario in dicionario:
14         if (palavraTexto == palavraDicionario):
15             bag_words[palavraDicionario] +=1
16
17 return (bag_words)
```

Listing 2: Match total

Metodologia

Relevância de
Textos com
Lógica Fuzzy

*Eric Calasans
de Barros,
José Genilson
da Silva Filho*

Capa

Introdução

Metodologia

Resultados

```
1 # Procurando match de radical
2 def mathRadical(texto, dicionario):
3     textoStemmer = [stemmer.stem(w.lower()) for w in
4                     texto] # Transforma as palavras do dicionario em
5                             radicais
6     dicStemmer = [stemmer.stem(w.lower()) for w in
7                   dicionario] # Transforma as palavras do texto em
8                               radicais
9     bag_words = {}
10    nMatch = {} # Palavras que nao concordam com o
11                radical
12
13    # Inicia o dicionario que sera usado para
14    # contabilizar os radicais e as palavras que nao
15    # tiveram match
16    for i in dicStemmer:
17        bag_words[i] = 0
18        nMatch[i] = 0
```

Listing 3: Match de radical

Metodologia

Relevância de
Textos com
Lógica Fuzzy

*Eric Calasans
de Barros,
José Genilson
da Silva Filho*

Capa

Introdução

Metodologia

Resultados

```
1  # Procura os radicais como feito no metodo
    anterior
2  for palavraTexto in textoStemmer:
3      for palavraDicionario in dicStemmer:
4          if (palavraTexto == palavraDicionario):
5              bag_words[palavraDicionario] += 1
6
7  # Procura as palavras que nao estao presente no
    texto e estao no dicionario
8  for palavraDicionario in dicStemmer:
9      if (not palavraDicionario in textoStemmer):
10         nMatch[palavraDicionario] += 1
11
12  return (bag_words , nMatch)
```

Listing 4: Match de radical(cont.)

Metodologia

Relevância de
Textos com
Lógica Fuzzy

*Eric Calasans
de Barros,
José Genilson
da Silva Filho*

Capa

Introdução

Metodologia

Resultados

```
1 #Relevancia de uma palavra
2 def relevancia():
3     #Inicializacoes
4     total = matchWord(word_features , palavras)
5     radical , nMatch = macthRadical(word_features ,
6                                     palavras)
7
8     relevancia_total_sum = 0
9     relevancia_radical_sum = 0
10    relevancia_nMatch_sum = 0
```

Listing 5: Relevância de cada palavra

Metodologia

Relevância de
Textos com
Lógica Fuzzy

*Eric Calasans
de Barros,
José Genilson
da Silva Filho*

Capa

Introdução

Metodologia

Resultados

```
1  for matchPalavra, key in enumerate(total):
2      if(relevancia_total_sum == 0):
3          relevancia_total_sum = 1
4          relevancia_total[key] = total[key]/
            relevancia_total_sum
5
6  for matchPalavra, key in enumerate(radical):
7      if(relevancia_radical_sum == 0):
8          relevancia_radical_sum = 1
9          relevancia_radical[key] = radical[key]/
            relevancia_radical_sum
```

Listing 6: Relevância de cada palavra(cont.)

Metodologia

Relevância de
Textos com
Lógica Fuzzy

*Eric Calasans
de Barros,
José Genilson
da Silva Filho*

Capa

Introdução

Metodologia

Resultados

```
1  for matchPalavra, key in enumerate(nMatch):  
2      if (relevancia_nMatch_sum == 0):  
3          relevancia_nMatch[key] = 0  
4      else:  
5          if (relevancia_nMatch_sum == 0):  
6              relevancia_nMatch_sum = 1  
7              relevancia_nMatch[key] = nMatch[key]/  
                relevancia_nMatch_sum
```

Listing 7: Relevância de cada palavra(cont.)

Metodologia

Relevância de
Textos com
Lógica Fuzzy

*Eric Calasans
de Barros,
José Genilson
da Silva Filho*

Capa

Introdução

Metodologia

Resultados

```
1  print(relevancia_total)
2  print(relevancia_radical)
3  print(relevancia_nMatch)
4
5  return(relevancia_total , relevancia_radical ,
        relevancia_nMatch)
```

Listing 8: Relevância de uma palavra(cont.)

Metodologia

Relevância de
Textos com
Lógica Fuzzy

*Eric Calasans
de Barros,
José Genilson
da Silva Filho*

Capa

Introdução

Metodologia

Resultados

```
1 def fuzzyRelText(relTotal=0, relRadical=0,  
2   relNoMatch=0):  
3     # Cria as variaveis fuzzy: Antecedentes e  
4     # Consequente  
5     # Antecedentes  
6     total = ctrl.Antecedent(np.arange(start=0, stop  
7       =1.1, step=0.1), 'Total')  
8     radical = ctrl.Antecedent(np.arange(start=0, stop  
9       =1.1, step=0.1), 'Radical')  
10    noMatch = ctrl.Antecedent(np.arange(start=0, stop  
11      =1.1, step=0.1), 'NoMatch')  
12  
13    #Consequente  
14    nivelRelevancia = ctrl.Consequent(np.arange(start  
15      =0.0, stop=1.1, step=0.1), 'Relevancia')
```

Listing 9: Lógica Fuzzy

Metodologia

Relevância de
Textos com
Lógica Fuzzy

*Eric Calasans
de Barros,
José Genilson
da Silva Filho*

Capa

Introdução

Metodologia

Resultados

```
1  #Funcoes
2  #Total
3  total['nRelevante'] = fuzzy.trimf(total.universe ,
4  [0, 0, 0.8])
5  total['relevante'] = fuzzy.trimf(total.universe ,
6  [0.2, 1, 1])
7  #Radical
8  radical['nRelevante'] = fuzzy.trimf(radical.
   universe, [0, 0, 0.8])
9  radical['relevante'] = fuzzy.trimf(radical.
   universe, [0.2, 1, 1])
```

Listing 10: Lógica Fuzzy

Metodologia

Relevância de
Textos com
Lógica Fuzzy

*Eric Calasans
de Barros,
José Genilson
da Silva Filho*

Capa

Introdução

Metodologia

Resultados

```
1  #NoMatch
2  noMatch[ 'nRelevante' ] = fuzzy.trimf(noMatch.
    universe , [0, 0, 0.8])
3  noMatch[ 'relevante' ] = fuzzy.trimf(noMatch.
    universe , [0.2, 1, 1])
4
5  #resultado
6  nivelRelevancia[ 'poucoRelevante' ] = fuzzy.trimf(
    nivelRelevancia.universe , [0, 0, 0.4])
7  nivelRelevancia[ 'relevante' ] = fuzzy.trimf(
    nivelRelevancia.universe , [0.1, 0.5, 0.9])
8  nivelRelevancia[ 'muitoRelevante' ] = fuzzy.trimf(
    nivelRelevancia.universe , [0.6, 1, 1])
```

Listing 11: Lógica Fuzzy(cont.)

Metodologia

Relevância de
Textos com
Lógica Fuzzy

*Eric Calasans
de Barros,
José Genilson
da Silva Filho*

Capa

Introdução

Metodologia

Resultados

```
1  # Regras
2  r1 = ctrl.Rule(antecedent=total['relevante'] &
3                radical['relevante'] & noMatch['nRelevante'],
4                consequent=nivelRelevancia['muitoRelevante'])
5  r2 = ctrl.Rule(antecedent=total['nRelevante'] &
6                radical['nRelevante'] & noMatch['relevante'],
7                consequent=nivelRelevancia['poucoRelevante'])
8  r3 = ctrl.Rule(antecedent=total['relevante'] &
9                radical['relevante'] & noMatch['nRelevante'],
10               consequent=nivelRelevancia['relevante']
11               ]%0.8)
12 r4 = ctrl.Rule(antecedent=total['relevante'] &
13               radical['relevante'] & noMatch['relevante'],
14               consequent=nivelRelevancia['relevante']
15               ]%0.6)
```

Listing 12: Lógica Fuzzy(cont.)

Metodologia

Relevância de
Textos com
Lógica Fuzzy

*Eric Calasans
de Barros,
José Genilson
da Silva Filho*

Capa

Introdução

Metodologia

Resultados

```
1  # Regras
2  r5 = ctrl.Rule(antecedent=total['relevante'] &
3                radical['nRelevante'] & noMatch['nRelevante'],
4                consequent=nivelRelevancia['relevante']
5                ]%0.7)
6  r6 = ctrl.Rule(antecedent=total['nRelevante'] &
7                radical['relevante'] & noMatch['nRelevante'],
8                consequent=nivelRelevancia['relevante']
9                ]%0.5)
10 r7 = ctrl.Rule(antecedent=total['nRelevante'] &
11               radical['nRelevante'] & noMatch['nRelevante'],
12               consequent=nivelRelevancia['poucoRelevante'])
```

Listing 13: Lógica Fuzzy(cont.)

Metodologia

Relevância de
Textos com
Lógica Fuzzy

*Eric Calasans
de Barros,
José Genilson
da Silva Filho*

Capa

Introdução

Metodologia

Resultados

```
1  #Cria maquina de inferencia
2  controleRelevancia = ctrl.ControlSystem([r1, r2,
3      r3, r4, r5, r6, r7])
4
5  #Prepara a simulacao
6  resultado = ctrl.ControlSystemSimulation(
7      control_system=controleRelevancia)
8
9  #Entrada de dados
10 resultado.input['Total'] = relTotal
    resultado.input['Radical'] = relRadical
    resultado.input['NoMatch'] = relNoMatch
```

Listing 14: Lógica Fuzzy(cont.)

Metodologia

Relevância de
Textos com
Lógica Fuzzy

*Eric Calasans
de Barros,
José Genilson
da Silva Filho*

Capa

Introdução

Metodologia

Resultados

```
1  #Defuzzificacao
2  resultado.compute()
3
4  #Retorna relevancia
5  return resultado.output[ 'Relevancia' ]
```

Listing 15: Lógica Fuzzy(cont.)

Metodologia

Relevância de
Textos com
Lógica Fuzzy

*Eric Calasans
de Barros,
José Genilson
da Silva Filho*

Capa

Introdução

Metodologia

Resultados

```
1 (total, radical, nMatch) = relevancia()
2 rele = 0
3
4 # A relevancia final do texto e calculada somando as
   relevancias de cada palavra do dicionario
   dividido pela quantidade
5 # de palavras no dicionario
6 for word, key in enumerate(total):
7     rele += fuzzyText.fuzzyRelText(total[key], radical
   [stemmer.stem(key)], nMatch[stemmer.stem(key)])
8
9 print(rele/len(total))
```

Listing 16: Lógica Fuzzy(cont.)

Resultados

Relevância de
Textos com
Lógica Fuzzy

*Eric Calasans
de Barros,
José Genilson
da Silva Filho*

Capa

Introdução

Metodologia

Resultados

Texto:

Há relatos de um esporte muito parecido com o futebol, embora usava-se muito a violência.

O Soule ou Harpastum era praticado na Idade Média por militares que se dividiam em duas equipes: atacantes e defensores. Era permitido usar socos, pontapés, rasteiras e outros golpes violentos. Há relatos que mostram a morte de alguns jogadores durante a partida.

Cada equipe era formada por 27 jogadores, onde grupos tinham funções diferentes no time: corredores, dianteiros, sacadores e guarda-redes. Na Itália Medieval apareceu um jogo denominado gioco del calcio. Era praticado em praças e os 27 jogadores de cada equipe deveriam levar a bola até os dois postes que ficavam nos dois cantos extremos da praça. A violência era muito comum, pois os participantes levavam para campo seus problemas causados, principalmente por questões sociais típicas da época medieval. O barulho, a desorganização e a violência eram tão grandes que o rei Eduardo II teve que decretar uma lei proibindo a prática do jogo, condenando a prisão os praticantes. Porém, o jogo não terminou, pois integrantes da nobreza criaram uma nova versão dele com regras que não permitiam a violência. Nesta nova versão, cerca de doze juízes deveriam fazer cumprir as regras do jogo.

Resultados

Relevância de
Textos com
Lógica Fuzzy

*Eric Calasans
de Barros,
José Genilson
da Silva Filho*

Capa

Introdução

Metodologia

Resultados

- **Dicionário:** jogador; futebol
- **Matches**