

SME0827 - Estruturas de Dados



Textos e Strings Aula 14



Professor: André C. P. L. F. de Carvalho, ICMC-USP
PAE: Moisés Rocha dos Santos
Monitora: Marília Costa Rosendo Silva



Mineração de textos

■ Uma das aplicações mais comuns

- Livros, artigos, blogs, formulários, mensagens em redes sociais,...

■ Dados não estruturados

- Textos precisam ser pré-processados antes de minerados

■ E estruturados

Dies ist ein Blindtext. An ihm lässt sich vieles über die Schrift ablesen, in der er gesetzt ist. Auf den ersten Blick wird der Grauwert der Schriftfläche sichtbar. Dann kann man prüfen, wie gut die Schrift zu lesen ist und wie sie auf den Leser wirkt. Dies ist ein Blindtext. An ihm lässt sich vieles über die Schrift ablesen, in der er gesetzt ist. Auf den ersten Blick wird der Grauwert der Schriftfläche sichtbar. Dann kann man prüfen, wie gut die Schrift zu lesen ist und wie sie auf den Leser wirkt.

© André de Carvalho - ICMC/USP

4



Aula de hoje

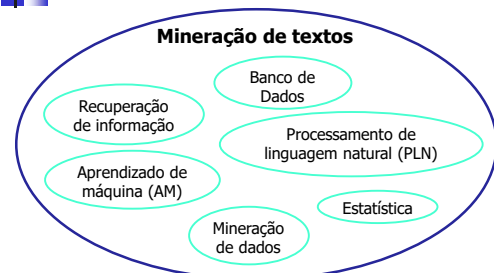
- Introdução
- Mineração de textos
- Textos e strings
- Operações para o tipo strings
- Indexação de strings
- Alteração de strings
- Funções e métodos para strings

© André de Carvalho - ICMC/USP

2



Área interdisciplinar



© André de Carvalho - ICMC/USP

5



Introdução

- Arquivos com textos são gerados com frequência crescente
 - Internet
 - Bioinformática
 - Processamento de documentos
- Várias tarefas atuais de programação são para manipular textos

© André de Carvalho - ICMC/USP

3



Tarefas

- Preditivas
 - Classificação e rotulação de textos
- Descritivas
 - Agrupamento de textos
- Extração de conceitos e entidades
- Análise de sentimentos e de opiniões
- Sumarização de documentos
- ...

© André de Carvalho - ICMC/USP

6



Mineração de texto

- Descritivo
 - Grupo de documentos semelhantes
 - Procure textos sobre questões e palavras semelhantes que frequentemente aparecem juntas
- Preditivo
 - Classificação de documentos em um ou mais tópicos
 - Análise de sentimentos e mineração de opinião



Mineração de textos

- Principais etapas
 - Aquisição de dados
 - Converta o texto em uma sequência de caracteres
 - Remoção de informações desnecessárias
 - Extração de características
 - Extrair recursos relevantes dos dados originais
 - Pré-processamento de dados
 - Indução do modelo
 - Avaliação e interpretação dos resultados



Aplicações

- Detecção de spam
- Análise de mídias sociais
 - Ex. engajamento de público-alvo
- Atendimento ao consumidor
 - Ex. utilização de *bots*
- Análise de processos jurídicos e de contratos



Extração de características

- Principais passos para extrair características de textos
 - Tokenização (define componentes)
 - Stemming (gera radicais)
 - Lematização (gera lemas)
 - Remoção de stop words (reduz dados)
 - Conversão para dados estruturados



Mensagens em redes sociais



Extração de características

Mensagens recebidas	Classe
I like my sister's birthday party	Família
I liked the company party	Trabalho
I am not bringing them from school	Família
I will talk and bring the contract	Trabalho
I talked to other companies	Trabalho
My wife is having contractions	Família

- Tokenização
 - Decompõe um texto em uma sequência de palavras (tokens)
 - Ignora espaços brancos e sinais de pontuação
 - Se uma palavra aparecer mais de uma vez no texto:
 - Ela aparecerá mais de uma vez na sequência de tokens
 - Bag of words (BOW)
 - Lista de tokens que aparecem no texto
 - Term frequency (TF)
 - Lista de tokens com frequência



Stemming

Tokens	Stems (radicais)
studied, studying, student, studies, study	studi, studi, student, studi, studi
miner, mining, mine	miner, mine, mine
vegetable, vegetarian, vegetate	veget, vegetarian, veget, veget
eating, ate, eats, eater	eat, ate, eat, eater

- Adota um critério comum para representar variações de tokens
- Evita um grande número de variações de tokens
 - Pode gerar um conjunto de dados muito esparsos
- Converte cada token para um radical
 - Algoritmos de Stemming
 - Porter (mais usado) e Snowball
 - Stem de "studied", "studying", "studies", "study": **studi**
 - Stem de "student": **student**

© André de Carvalho - ICMC/USP

13



Remoção de stop words

Stems após remoção de stop words	Classe
sister, birthday, parti	Família
compani, parti	Trabalho
bring, school	Família
talk, bring, contract	Trabalho
talk, compani	Trabalho
wife, contract	Família

- Reduz ainda mais o número de stems removendo stop words
 - Adjéctivos (good, bad, large...)
 - Adverbios (fast, nicely, not...)
 - Artigos (a, an, the)
 - Negações (none, not, never...)
 - Pronomes (I, he, my, his, yours, ours...)
 - Preposições (at, by, for, from, in, to...)
 - Conjunções (and, but, or, with...)
 - Verbos freq. (are, be, is, was, has...)
 - Qualificadores (a little, less, more, very, yet...)

© André de Carvalho - ICMC/USP

16



Stemming

Mensagens recebidas após stemming	Class
I, like, my, sister, birthday, parti	Família
I, like, the, compani, parti	Trabalho
I, am, not, bring, them, from, school	Família
I, will, talk, and, bring, the, contract	Trabalho
I, talk, to, other, compani	Trabalho
My, wife, is, have, contract	Família

- Adota um critério comum para representar variações de tokens
- Evita um grande número de variações de tokens
 - Pode gerar um conjunto de dados muito esparsos
- Converte cada token para um radical
 - Algoritmos de Stemming
 - Porter (mais usado) e Snowball
 - Stem de "studied", "studying", "studies", "study": **studi**
 - Stem de "student": **student**

© André de Carvalho - ICMC/USP

14



Remoção de stop words

Stems após remoção de stop words	Classe
sister, birthday, parti	Família
compani, parti	Trabalho
bring, school	Família
talk, bring, contract	Trabalho
talk, compani	Trabalho
wife, contract	Família

- Decisão de quando parar de remover palavras depende da aplicação
 - Ex.: presença de adjéctivos e negações é importante para mineração de opiniões
- Reduz a esparsidade da tabela com os dados estruturados

© André de Carvalho - ICMC/USP

17



Lematização

Mensagens recebidas	Classe
i like my sister birthday party	Família
i like the company party	Trabalho
i be not bring them from school	Família
i will talk and bring the contract	Trabalho
i talk to other company	Trabalho
my wife be have contraction	Família

- Variação mais elaborada de stemming
 - Usa um vocabulário e leva em conta gramática da linguagem
 - Faz uma análise morfológica
- Retorna o formato de uma palavra em dicionários, chamado Lema
- Mais preciso, porém mais lento que stemming

© André de Carvalho - ICMC/USP

15




Conversão para dados estruturados

- Cria uma tabela com valores binários (presença de um stem no texto) ou quantitativos (frequência de um stem no texto)

birthday	bring	company	party	school	sister	talk	wife	Classe
1	0	0	1	0	1	0	0	Família
0	0	1	1	0	0	0	0	Trabalho
0	1	0	0	1	0	0	0	Família
0	1	0	0	0	0	1	0	Trabalho
0	0	1	0	0	0	1	0	Trabalho
0	0	0	0	0	0	0	1	Família

© André de Carvalho - ICMC/USP

18



© André de Carvalho - ICMC/USP 19

Fatiamento de strings

- *Substrings* podem ser indicados pelo uso da notação de fatias
 - Valores inicial e final separados por ":"
 - Fatia não inclui caracter do último índice
 - Ex.: `palavra[0:2]` # não inclui `palavra[2]`
 - Valores extremos podem ser negativos
 - Ex.: `palavra[1:-1]` # não inclui `palavra[0]` nem `palavra[-1]`
 - Obs.: `palavra[-1]` indexa o último caracter de `palavra`

© André de Carvalho - ICMC/USP 22

Operadores para strings

- Python usa o tipo string para manipular textos
- Strings podem ser:
 - Comparados com operadores `==`, `!=`, `<`, `>`, `<=` e `>=`
 - Concatenados com o operador `+`
 - Repetidos com o operador `*`

© André de Carvalho - ICMC/USP 20

Fatiamento de strings

- Valores extremos podem ser omitidos
 - São usados valores default (0 e posição do último elemento)
 - Mas o sinal ":" não (pois é obrigatório)
- É possível usar um terceiro argumento
 - Passo do fatiamento
 - Por default, igual a 1
 - Ex.: `palavra [0:2:2]`

© André de Carvalho - ICMC/USP 23

Indexação de strings

- *Strings* podem ser indexados
 - Primeiro elemento possui índice 0
 - Não existe um tipo *caracter*
 - *Caracter* é um *string* de tamanho 1
 - É possível utilizar intervalos para indexar um ou mais *caracteres*
 - *Substrings*

© André de Carvalho - ICMC/USP 21

Alteração de Strings

- Python não permite mudar o valor dos elementos de um string
 - Tipo imutável
 - Diferente da linguagem C
- Mas é fácil e eficiente combinar partes de strings para criar um novo string

© André de Carvalho - ICMC/USP 24



Métodos gerais

- Não restritos a strings
- *len (tex)*
 - Retorna o tamanho (número de caracteres) do string *tex*
- *subtex in tex*
 - Verifica se substring *subtex* pertence ao (ocorre no) string *tex*
 - *Substring pode ter apenas um caracter*



Conclusão

- Introdução
- Mineração de textos
- Operações para o tipo string
- Indexação de strings
- Alteração de strings
- Métodos para strings



Métodos para strings

- *tex.find (subtex)*
 - Retorna posição de primeira ocorrência do *substring subtex* no *string tex*
- *tex.count (subtex)*
 - Conta número de vezes que *subtex* ocorre no *string tex*
- *tex.replace (strvelho, strnovo)*
 - Troca em *tex* ocorrência de *strvelho* por *strnovo*



Perguntas



Métodos para strings

- *tex.strip (subtex)*
 - Remove substring *subtex* do início e final de um string *tex*
 - Parâmetro do tipo *string* (*string* a ser removido)
 - Não modifica o string *tex*
 - Gera um novo *string*