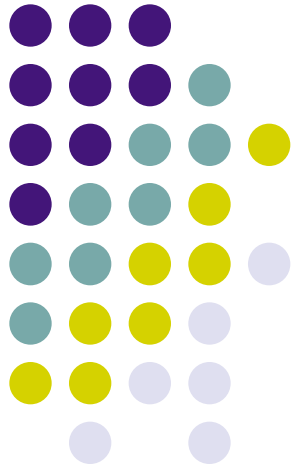
Interpretação de Linguagem Natural

Aula 10



O conteúdo destes slides foi adaptado de:

- **Curso "Da Linguagem Natural a Informação" – Prof. Emerson Cabrera Paraiso (PPGIa/PUCPR).**
- **"Natural Language Processing" course – Prof. Thamar Solorio (University of Houston).**
- **"Speech and Language Processing", Jurafsky, D., Martin, J.; 3a Edição (2018).**

Pontifícia Universidade Católica do Paraná (PUCPR)

Bacharelado em Ciência da Computação – 4º Período

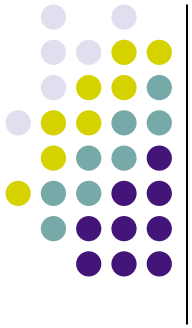
gregory.puppi@pucpr.br

Expressões Regulares (REGEX) (Aula Anterior)

- Expressões Regulares (Regular Expressions): linguagem formal para especificar cadeias de caracteres (strings).
- É uma das formas mais básicas de processar um texto.
- Permite a especificação de padrões utilizados na busca de strings (ou substrings) em textos.
- Após a construção do padrão, um motor faz a análise léxica e sintática do texto-alvo e indica as ocorrências das strings encontradas a partir do padrão indicado.
- Trata-se de ferramenta muito utilizada na recuperação da informação.
- Define padrões para o processo de tokenização.

Exemplos de REGEX (Aula Anterior)

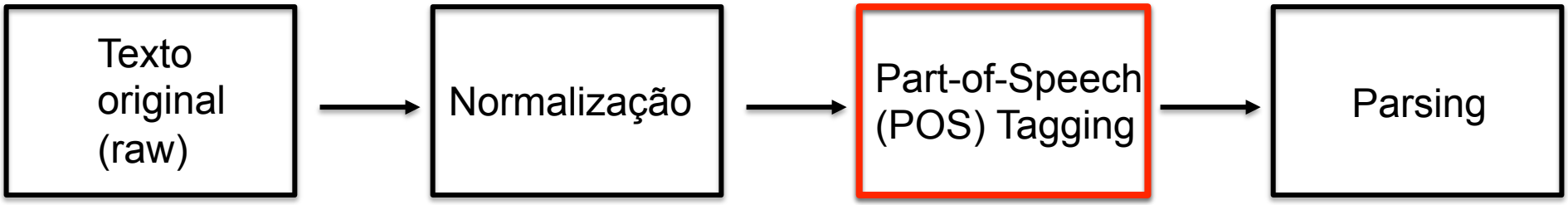
- Verificar a presença do http:// ou https://
  - ^(http:\/\/www\.|https:\/\/www\.|http:\/\/|https:\/\/)?[a-z0-9]+([\-\.]{1}[a-z0-9]+)*\.[a-z]{2,5}(:[0-9]{1,5})?(\/.*)?$

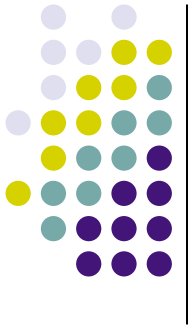


Tipos de Erros (por Jurafsky) (Aula Anterior)

- No exemplo, dois tipos de erros surgiram:
  - 1) Retornar strings indesejadas (“parado”): falso positivo
  - 2) Não retornar strings desejadas (“Para”): falso negativo

Exemplo de Etapas Típicas do PLN (Aula Anterior)





Part-of-Speech (POS) Tagging (Aula Anterior)

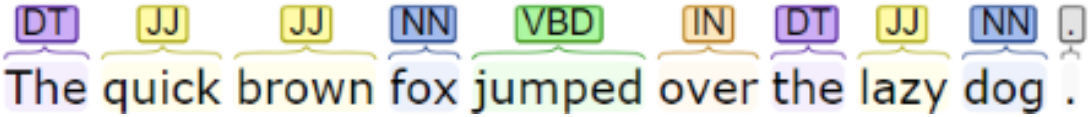
- Análise morfológica.
- Processo de atribuir uma part-of-speech (classe gramatical) para cada palavra num corpus. (Jurafsky and Martin)
- Exemplos de classes gramaticais:
  - Substantivos, verbos, pronomes, adjetivos, advérbios, etc.

Conjunto de POS Tags (Aula Anterior)

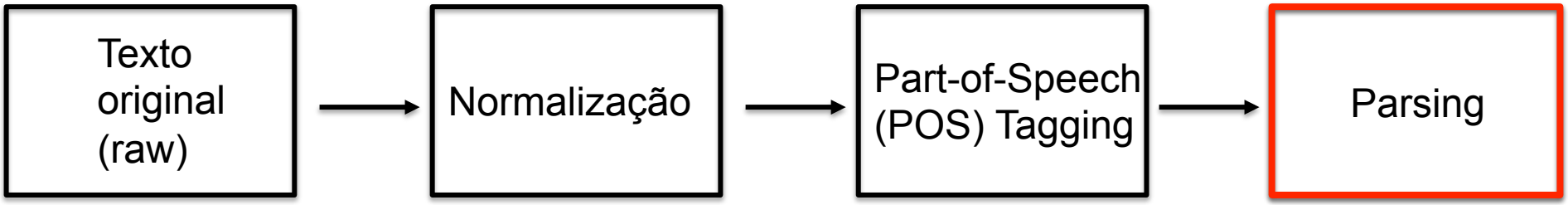
Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two, three*	TO	"to"	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential 'there'	*there*	VB	verb, base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb, past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb, gerund	*eating*
JJ	adjective	*yellow*	VBN	verb, past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb, non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb, 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, singular	*IBM*	$	dollar sign	*$*
NNPS	proper noun, plural	*Carolinas*	#	pound sign	#
PDT	predeterminer	*all, both*	"	left quote	' or "
POS	possessive ending	*'s*	"	right quote	' or "
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one's*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

Part-of-Speech (POS) Tagging (Aula Anterior)

- Exemplo de POS tagging realizado com StanfordNLP.



Exemplo de Etapas Típicas do PLN (Aula Anterior)

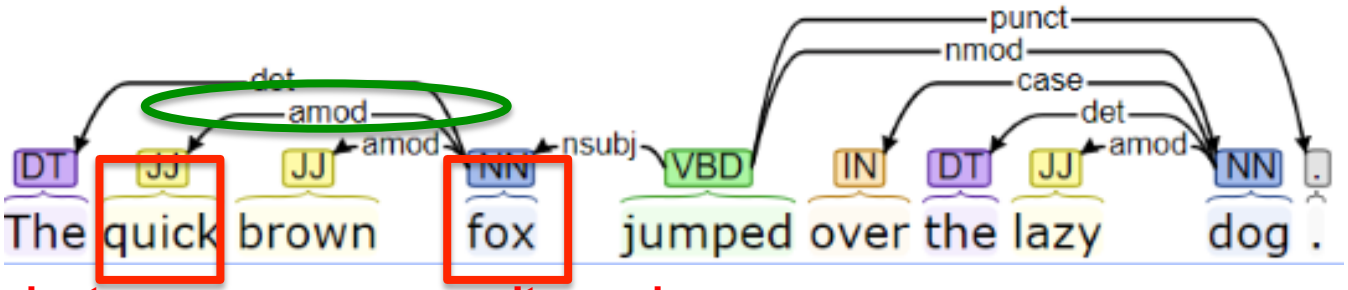


Parsing (Aula Anterior)

- Existem diferentes tipos
  - Foco: **parser de dependência**.

Parsing (Aula Anterior)

- Parser de dependência (*dependency parser*)
  - Representação da estrutura sintática: consiste de itens léxicos conectados por relações binárias (dependências).



Item dependente **Item raiz**

A flecha conecta um item léxico chamado raiz (*head*, *governor*) com um item léxico denominado dependente (*subordinate*). Normalmente as dependências formam uma árvore.

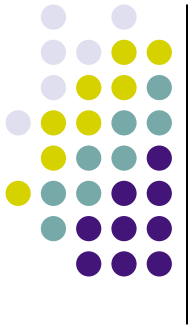
Parser de Dependência com StanfordNLP (Aula Anterior)

Índices na entrada: 1 2 3 4 5 6 7 8 9 10

```
========== Dependency Parser ==========
Sentence: " The quick brown fox jumped over the lazy dog. "

('The', '4', 'det')
('quick', '4', 'amod')
('brown', '4', 'amod')
('fox', '5', 'nsubj')
('jumped', '0', 'root')
('over', '9', 'case')
('the', '9', 'det')
('lazy', '9', 'amod')
('dog', '5', 'obl')
('.', '5', 'punct')
```

Dependências são tuplas:
(item dependente,
índice do item raiz na frase,
nome da relação)



Plano de Aula

- Representação Vetorial de Textos
  - Bag-of-words
  - Matriz Termo-Documento
  - TF-IDF
- Similaridade entre Documentos

Representação Vetorial de Textos

- Definição: trata-se de conversão da representação textual (strings) de um corpus para uma representação numérica (vetor).
- Esta operação é necessária como etapa inicial ao processo de classificação ou recuperação da informação, por exemplo.
- Em outras palavras, trata-se de um processo equivalente a “obtenção de características” do texto/corpus.
- O método mais conhecido para realizar esta tarefa é o bag-of-words.

Bag-of-words (BoW)

- O modelo bag-of-words (BoW) propõe uma maneira de representar as características textuais de documentos em vetores numéricos.
- O BoW é baseado na frequência de palavras nos textos (histograma de palavras).
- A ideia da “sacola” de palavras vem do fato de que a ordem das palavras ou a estrutura do texto não é levado em consideração no processo.
- Todo o corpus pode ser chamado de “lista de BoW”.

BoW

- **Primeira etapa:** encontrar todas as ocorrências de uma palavra (ou termo), o que chamaremos de definição do vocabulário.

- Dado o seguinte corpus extraído de (https://machinelearningmastery.com/gentle-introduction-bag-words- model/):
  - “It was the best of times,”
  - “it was the worst of times,”
  - “it was the age of wisdom,”
  - “it was the age of foolishness.”

- Cada linha é tratada como um documento. O vocabulário seria então formado por 10 palavras:
  - “it”, “was”, “the”, “best”, “of”, “times”, “worst”, “age”, “wisdom”, “foolishness”

BoW

- Perceba que nenhuma operação básica de modificação no texto foi realizada (por exemplo, extração de stopwords).

BoW

- **Segunda etapa:** criação dos vetores de documentos. Os vetores terão comprimento de 10 posições, visto que o vocabulário tem comprimento C = 10. Para cada texto, indicar a ocorrência (e a quantidade) para cada termo.

- Dado o vocabulário, cada palavra representa uma posição no vetor:
  - {“it”, “was”, “the”, “best”, “of”, “times”, “worst”, “age”, “wisdom”, “foolishness”}

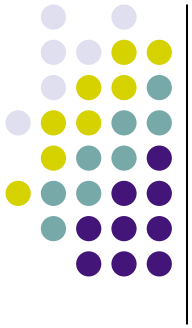
- Vetores:
  - [1, 1, 1, 1, 1, 1, 0, 0, 0, 0]
  - [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]
  - [1, 1, 1, 0, 1, 0, 0, 1, 1, 0]
  - [1, 1, 1, 0, 1, 0, 0, 1, 0, 1]

Outro Exemplo

- Corpus:
  - “John likes to watch movies. Mary likes movies too.”
  - “John also likes to watch football games.”

- Vocabulário após a extração de stopwords {“also”, “to”, “too”}:
  - {“John”, “likes”, “watch”, “movies”, “Mary”, “football”, “games”}

- Vetores:
  - [1, 2, 1, 2, 1, 0, 0]
  - [1, 1, 1, 0, 0, 1, 1]



Plano de Aula

- Representação Vetorial de Textos
  - Bag-of-words
  - Matriz Termo-Documento
  - TF-IDF
- Similaridade entre Documentos

Matriz Termo-Documento

- O conjunto de todos os vetores forma a matriz Termo-Documento.
- Para a coleção de documentos a seguir:
  - d1 = “O carro branco é bonito. O carro é novo.”
  - d2 = “Comprei um carro branco bonito.”
  - d3 = “Comprei um novo carro.”
  - d4 = “Todos precisamos de um carro.”
    - A matriz equivalente é construída (sem stopwords):

	carro	branco	bonito	comprei	novo	precisamos
d_1	2	1	1	0	1	0
d_2	1	1	1	1	0	0
d_3	1	0	0	1	1	0
d_4	1	0	0	0	0	1

Detalhes da Utilização do BoW

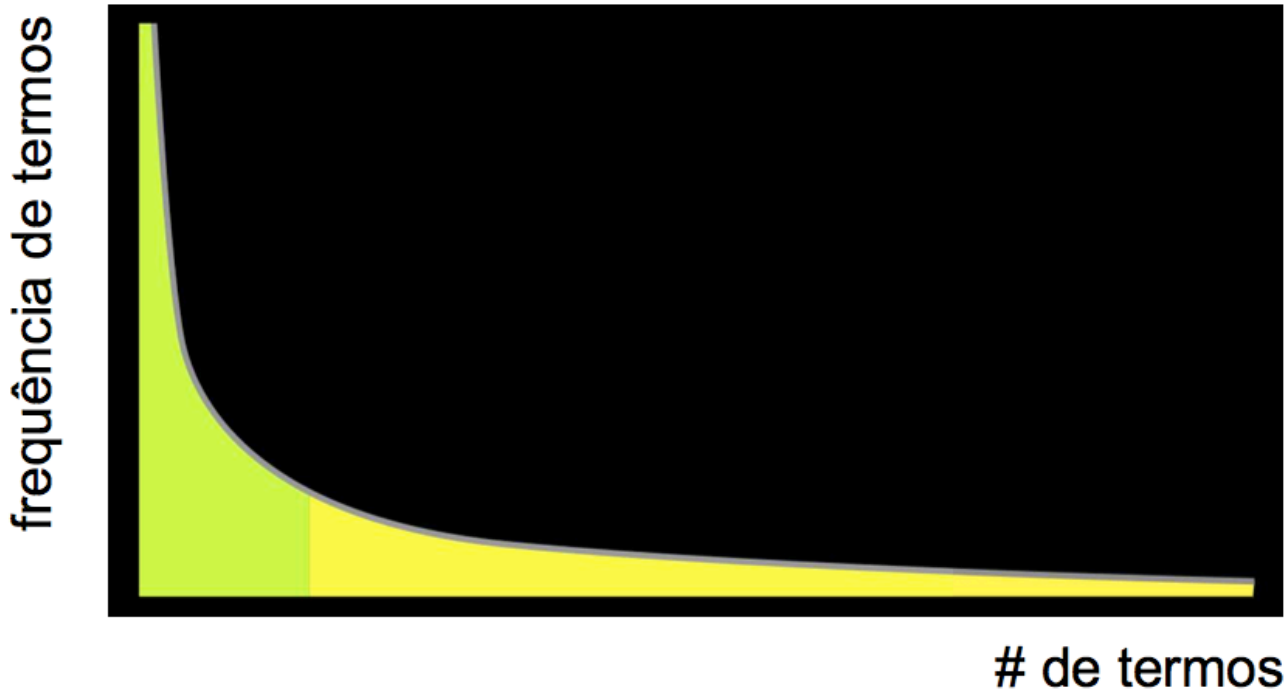
- A matriz gerada pelos N documentos (textos) é geralmente esparsa (vários 0s ao logo das colunas - Lei de Zipf) e de alta dimensionalidade.

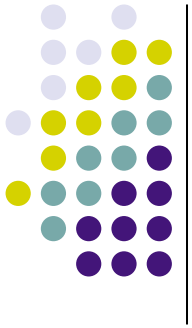
- Formas de reduzir a dimensão:
  - Retirada de stopwords;
  - Lematização;
  - Uso do n-gram (múltiplas palavras por token).

- O fato de uma palavra ter alta frequência não necessariamente significa que trata-se de um termo importante. Por exemplo: artigos (‘o’, ‘a’, ...) tem a tendência de ocorrer com frequência em textos.

Distribuição dos Termos

- A distribuição dos termos em uma coleção de documentos segue a lei de Zipf e a distribuição de cauda longa (long-tailed):
  - A maior parte do vocabulário tem baixa frequência.





Plano de Aula

- Representação Vetorial de Textos
  - Bag-of-words
  - Matriz Termo-Documento
  - TF-IDF
- Similaridade entre Documentos

TF-IDF

- Definição: o TF-IDF (term frequency–inverse document frequency) é usado para medir a importância de um termo em um documento presente em uma coleção de documentos. ((JONES, 1972); (SALTON; BUCKLEY, 1988))

- O valor TF-IDF de uma palavra aumenta proporcionalmente à medida que aumenta o número de ocorrências dela em um documento. Porém, este valor é relativizado pela frequência da palavra no corpus. Resumindo:
  - 1) Quanto mais frequentemente um termo ocorre em um documento, mais representativo ele é para o conteúdo, e;
  - 2) Quanto mais documentos o termo ocorre, menos discriminativo ele é.

TF-IDF

- “TF-IDF é comumente usado em Recuperação de Informação para comparar um vetor de consulta com um vetor de um documento de texto, usando uma função de similaridade ou distância, como a função cosseno”. (SOUCY; MINEAU, 2005)

- Para computar o TF-IDF vamos trabalhar com o seguinte corpus:
  - d1 = “O carro branco está na rodovia.”
  - d2 = “O caminhão branco parou na garagem.”

	o	carro	branco	está	na	rodovia	caminhão	parou	garagem
d1	1	1	1	1	1	1	0	0	0
d2	1	0	1	0	1	0	1	1	1

Formulação Matemática – tf

- Matematicamente, TF-IDF (*term frequency–inverse document frequency*) pode ser computado como:
  - termo-frequência (tf): nos fornece a frequência de cada termo em um documento do corpus.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

    - Sendo $n_{i,j}$, a frequência de um termo i num documento j.

Formulação Matemática – tf

- Para o corpus:
  - tf(“carro”, d1) = 1/6 = 0.167
  - tf(“carro”, d2) = 0/6 = 0
  - tf(“branco”, d1) = 1/6 = 0.167
  - tf(“branco”, d2) = 1/6 = 0.167

Formulação Matemática – idf

- Cálculo do IDF (Inverse Document Frequency): permite computar o peso de cada palavra na coleção de documentos. Palavras que ocorrem mais raramente tem maior IDF.

$$idf(w) = log(\frac{N}{df_t})$$

- Sendo:
  - N = número total de documentos do corpus;
  - df_t = número de documentos contendo o termo t;

Formulação Matemática – idf

- Para o corpus:
  - idf (“carro”) = log(2/1) = 0.3
  - idf (“branco”) = log(2/2) = 0

Formulação Final

- O cálculo do TD-IDF seria então o produto de ambas equações:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

  - Sendo:
    - $w_{i,j}$ = trata-se do TF-IDF de um termo *i* em um documento *j*.

Exemplo

- Para o corpus:
  - tf-idf (“carro”, d1) = 0.167 x 0.3 = 0.051
  - tf-idf(“carro”,d2)=0x0.3=0
  - tf-idf (“branco”, d1) = 0.167 x 0 = 0
  - tf-idf (“branco”, d2) = 0.167 x 0 = 0

- Conclusões:
  - tf-idf de palavras em comum nos documentos é zero, ou seja, não são palavras significantes na discriminação dos textos;
  - if-idf de “carro” é diferente de zero, o que significa que esta palavra tem mais importância na coleção de documentos.

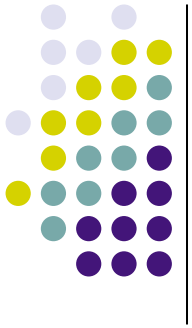
Outras Alternativas

- Caso você busque uma forma alternativa para representar os documentos, dê uma olhada neste artigo, é uma dentre várias possibilidades:
  - https://arxiv.org/pdf/1301.6770.pdf

Trabalho 7 (Parte I)

- 1) Dado o corpus:
  - d1 = “O rato roeu a roupa do rei de Roma.”
  - d2 = “Nenhum rato rói a roupa do rei de Roma sem punição.”
  - d3 = “A rota de fuga do rato foi rápida.”

Implementar um programa em Python para computar o TF-IDF de cada termo.



Plano de Aula

- Representação Vetorial de Textos
  - Bag-of-words
  - Matriz Termo-Documento
  - TF-IDF
- Similaridade entre Documentos

Similaridade entre Documentos

- Dada a introdução ao trabalho com o modelo de espaço vetorial (vector space model), onde documentos são representados como coleções (vetores) de valores (índices, frequência, etc.), podemos utilizá-los em diferentes aplicações.

- Vamos utilizar o espaço vetorial para identificar qual documento *D* está mais "próximo" de um vetor de consulta *Q* (a query *Q* será então considerada como um documento).

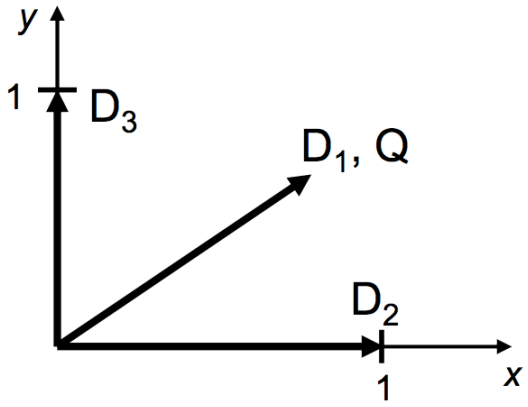
- Neste caso uma medida de similaridade pode ser usada para calcular a distância entre os vetores (documentos).

- A forma tradicional de medir a distância entre dois vetores é por meio da medida do ângulo entre ambos.

- O ângulo é computado pelo produto interno entre os vetores.

Similaridade entre Vetores

- No exemplo, cada documento é composto por dois termos.
- ‘A’ e ‘I’ são termos válidos.
  - Documentos:
    - $D_1 = \{A, I\}$
    - $D_2 = \{A\}$
    - $D_3 = \{I\}$
    - $Q = \{A, I\}$
  - Vetor de documentos:
    - $D_1 = [1, 1]$
    - $D_2 = [1, 0]$
    - $D_3 = [0, 1]$
    - $Q = [1, 1]$



Coeficiente de Similaridade (SC)

- Para o cálculo da similaridade entre os documentos temos diferentes abordagens. Uma das mais simples é calcular o produto dos vetores. Assume-se que o comprimento do vetor que representa a query Q é igual ao comprimento dos vetores dos documentos da coleção.

$$SC(Q, D_i) = \sum_{j=1}^{t} w_{qj} \times d_{ij}$$

- Sendo:
  - d_{ij} = é o peso do termo *j* do documento *I*
  - w_{qj} =é o peso do termo *j* da query *Q*

Exemplo

- Assumindo o seguinte corpus de documentos (Grossman and Frieder, 2004):
  - D1 = “Shipment of gold damaged in a fire.”
  - D2 = “Delivery of silver arrived in a silver truck.”
  - D3 = “Shipment of gold arrived in a truck.”

- e a query:
  - Q = “gold silver truck.”

- Temos então 3 documentos e 11 termos na coleção.

	a	arrived	damaged	delivery	fire	gold	in	of	shipment	silver	truck
d1	0	0	0.477	0	0.477	0.176	0	0	0.176	0	0
d2	0	0.176	0	0.477	0	0	0	0	0	0.477	0.176
d3	0	0.176	0	0	0	0.176	0	0	0.176	0	0.176
Q	0	0	0	0	0	0.176	0	0	0	0.477	0.176

Exemplo

- Após o cálculo dos pesos para cada documento, computamos a similaridade *SC* da query *Q* em relação à cada documento D_i.
  - $SC(Q, D_1)$ = (0 * 0) + (0 * 0) + (0 * 0.477) + ... + (0.477 * 0) + (0.176 * 0) = 0.031
  - $SC(Q, D_2)$ = (0.954 * 0.477) + (0.176 * 0.176) = 0.486
  - $SC(Q, D_3)$ = (0.176 * 0.176) + (0.176 * 0.176) = 0.062

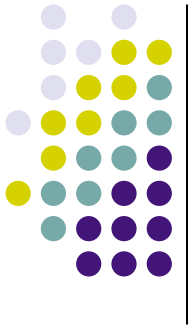
- Assim, o documento mais próximo à query *Q* seria D_2, depois D_3 e D_1.

Trabalho 7 (Parte II)

- 1) Dado o corpus “30NoticiasCurtas”, computar o tf-idf das seguintes palavras:
  - “Brasil”
  - “mortos”
  - “governo”

- 2) Dado o corpus “30NoticiasCurtas”, comprovar lei de Gorge Zipf. Traçar um gráfico para visualização dos resultados.

- 3) Será que o conteúdo desta aula pode ser usado na extração de termos relevantes:
  - Faça um teste disso:
    - https://app.monkeylearn.com/main/extractors/ex_y7BPYzNG/tab/de/



Dúvidas?