



# Formação Inteligência Artificial

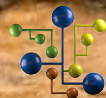


# Introdução à Inteligência Artificial



# Processamento de Linguagem Natural





# Processamento de Linguagem Natural

Há duas razões principais pelas quais queremos que nossos agentes de computador sejam capazes de processar linguagens naturais: primeiro, para se comunicar com os seres humanos e, segundo, para adquirir informação a partir da linguagem escrita.





# Processamento de Linguagem Natural

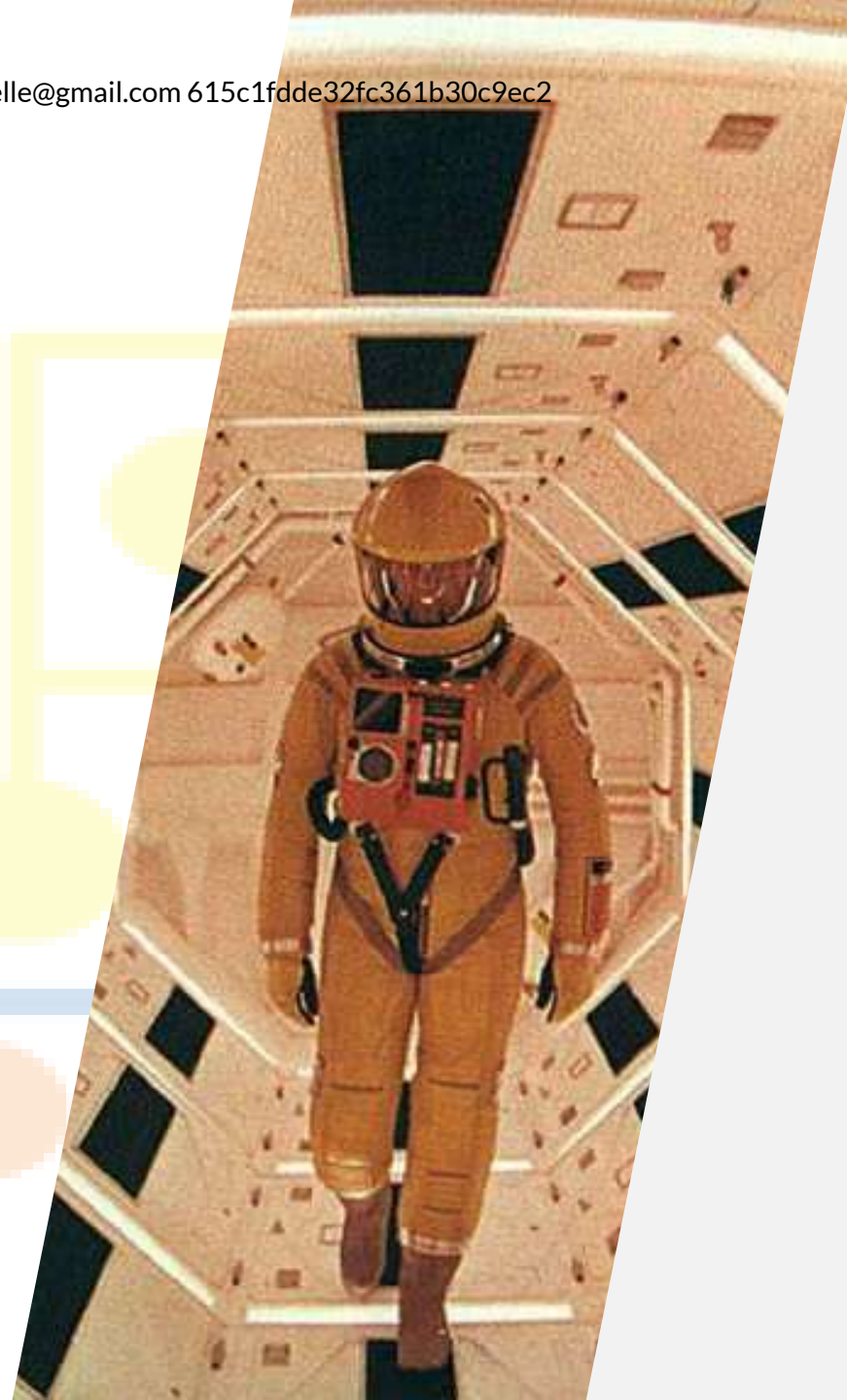
Um agente que deseja **adquirir conhecimento** precisa entender (pelo menos parcialmente) a ambígua e confusa linguagem que os seres humanos usam.





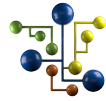
# Processamento de Linguagem Natural

Processamento de linguagem natural também inclui a capacidade de extrair insights de dados contidos em e-mails, vídeos e outros materiais não estruturados.





# O Que é Processamento de Linguagem Natural?

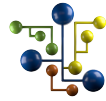


# O Que é Processamento de Linguagem Natural?

Processamento de Linguagem Natural (PLN) é uma subárea da ciência da computação, inteligência artificial e da linguística que estuda os problemas da geração e compreensão automática de línguas humanas naturais.







# O Que é Processamento de Linguagem Natural?

A história do PLN começou na década de 1950, quando Alan Turing publicou o artigo "**Computing Machinery and Intelligence**", que propunha o que agora é chamado de teste de Turing como critério de inteligência.





# O Que é Processamento de Linguagem Natural?

A marcação de partes da fala ([part-of-speech tagging](#)) introduziu o uso de [modelos ocultos de Markov](#) para o PLN e, cada vez mais, a pesquisa se concentrava em [modelos estatísticos](#), que tomam decisões suaves e probabilísticas baseadas na atribuição de pesos reais aos recursos que compõem dados de entrada.





# O Que é Processamento de Linguagem Natural?

Muitos dos sucessos iniciais notáveis ocorreram no campo da tradução automática, devido especialmente ao trabalho de pesquisa da IBM, que desenvolveu modelos estatísticos mais elaborados.







# O Que é Processamento de Linguagem Natural?

Pesquisas recentes têm se concentrado cada vez mais em algoritmos de aprendizagem semi-supervisionados e sem supervisão.



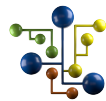




# O Que é Processamento de Linguagem Natural?

Um corpus (plural "[corpora](#)") é um conjunto de documentos (ou frases individuais) que foram marcados (labels) com os valores corretos a serem aprendidos.

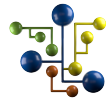




# O Que é Processamento de Linguagem Natural?

Atualmente, Deep Learning é o que há de mais avançado em processamento de linguagem natural.





# O Que é Processamento de Linguagem Natural?

Os procedimentos de aprendizagem de máquina podem fazer uso de algoritmos de inferência estatística para produzir modelos que são robustos a entradas não familiares (por exemplo, contendo palavras ou estruturas que não foram vistas antes) e a entradas errôneas (por exemplo, com palavras incorretamente omitidas).





# Aplicações do Processamento de Linguagem Natural

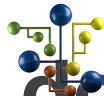




# Aplicações do Processamento de Linguagem Natural

## Sumarização Automática

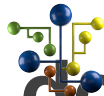




# Aplicações do Processamento de Linguagem Natural

## Análise de Discurso





# Aplicações do Processamento de Linguagem Natural

## Segmentação Morfológica

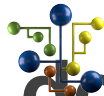


# Aplicações do Processamento de Linguagem Natural

## Análise Sintática (Parsing)







# Aplicações do Processamento de Linguagem Natural

## Análise Morfológica e Segmentação de Palavras





# Aplicações do Processamento de Linguagem Natural

## Geração de Linguagem Natural



# Aplicações do Processamento de Linguagem Natural

## Compreensão da Linguagem Natural



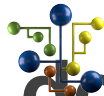


# Aplicações do Processamento de Linguagem Natural

## Análise Morfológica e Reconhecimento de Tópicos



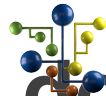




# Aplicações do Processamento de Linguagem Natural

## Marcação de Classe Gramatical





# Aplicações do Processamento de Linguagem Natural

## Máquina de Tradução





# Aplicações do Processamento de Linguagem Natural

## Reconhecimento de Entidades Nomeadas

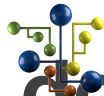




# Aplicações do Processamento de Linguagem Natural

## Reconhecimento Ótico de Caracteres (OCR)





# Aplicações do Processamento de Linguagem Natural

Respostas a Perguntas

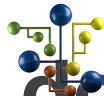




# Aplicações do Processamento de Linguagem Natural

## Extração de Relacionamento





# Aplicações do Processamento de Linguagem Natural

## Quebra de Frases

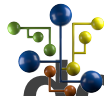




# Aplicações do Processamento de Linguagem Natural

## Análise de Subjetividade





# Aplicações do Processamento de Linguagem Natural

## Reconhecimento da Fala





# Aplicações do Processamento de Linguagem Natural

## Segmentação da Fala

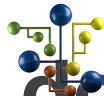






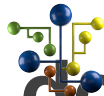
# Aplicações do Processamento de Linguagem Natural

Recuperação de Informação



# Aplicações do Processamento de Linguagem Natural

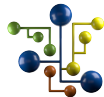
## Extração de Informação



# Aplicações do Processamento de Linguagem Natural

E esses são apenas alguns exemplos!!!

# Modelos de Linguagem



# Modelos de Linguagem

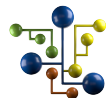
As linguagens formais, tais como as linguagens de programação Java ou Python, têm modelos de linguagem precisamente definidos.

Uma **linguagem** pode ser definida como um conjunto de sequências.

“**print (2 + 2)**” é um programa válido na linguagem Python, enquanto “**2) + (2 print**” não é







# Modelos de Linguagem



“Não ser convidado é triste”



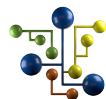


# Modelos de Linguagem



$$P(S = \text{palavras})$$





# Modelos de Linguagem



"Ele viu o banco"



# Modelo n-grama e Cadeia de Markov



# Modelo n-grama e Cadeia de Markov

Em última análise, um texto escrito é composto por **caracteres** — letras, dígitos, pontuação e espaços (e caracteres mais exóticos em alguns outros idiomas).

Escrevemos  $P(c1:cn)$  para a probabilidade de uma sequência de  $N$  caracteres, de  $c1$  até  $cn$ .

Uma sequência de símbolos escritos de comprimento  $n$  é chamado de  $n$ -grama (da raiz grega para escrita ou letras), com o caso especial de 1-grama para “unigrama”, 2-gramas para “bigrama” e 3-gramas pra “trigrama”







# Modelo n-grama e Cadeia de Markov

Um modelo de  $n$ -grama é definido como uma **cadeia de Markov** de ordem  $n - 1$

$$P(c_i | c_{1:i-1}) = P(c_i | c_{i-2:i-1})$$





# Modelo n-grama e Cadeia de Markov

Podemos definir a probabilidade de uma sequência de caracteres  $P(c_1:N)$  sob o modelo do trigrama primeiro por fatoração com a regra de cadeia e, em seguida, utilizando a hipótese de Markov:

$$P(c_{1:N}) = \prod_{i=1}^N P(c_i | c_{1:i-1}) = \prod_{i=1}^N P(c_i | c_{i-2:i-1})$$

Para um modelo de caracteres de trigrama em uma linguagem com 100 caracteres,  $P(C_i | C_{i-2:i-1})$  tem um milhão de entradas e pode ser estimado com precisão, contando as sequências de caracteres em um corpo de texto de 10 milhões de caracteres ou mais





# Modelo n-grama e Cadeia de Markov

O que podemos fazer com os modelos de caracteres de  $n$ -gramas?



# Modelo n-grama e Cadeia de Markov

Uma abordagem para identificação de idioma é primeiro construir um modelo de caracteres de trigrama de cada idioma candidato,  $P(c_i | c_{i-2:i-1}, \ell)$ , onde a variável  $\ell$  estende-se entre os idiomas.

$P(\text{Texto} | \text{Linguagem})$

$$\begin{aligned}\ell^* &= \operatorname{argmax}_{\ell} P(\ell | c_{1:N}) \\ &= \operatorname{argmax}_{\ell} P(\ell) P(c_{1:N} | \ell) \\ &= \operatorname{argmax}_{\ell} P(\ell) \prod_{i=1}^N P(c_i | c_{i-2:i-1}, \ell)\end{aligned}$$





# Modelo n-grama e Cadeia de Markov

Outras tarefas para modelos de caracteres incluem a correção de ortografia, classificação de gênero e o chamado reconhecimento de entidade.



# Modelo n-grama e Cadeia de Markov

A principal complicação de modelos *n*-grama é que o *corpus* de treinamento fornece apenas uma estimativa da distribuição de probabilidade verdadeira.

Para sequências de caracteres comuns, como “ht” qualquer *corpus* em inglês dará uma boa estimativa

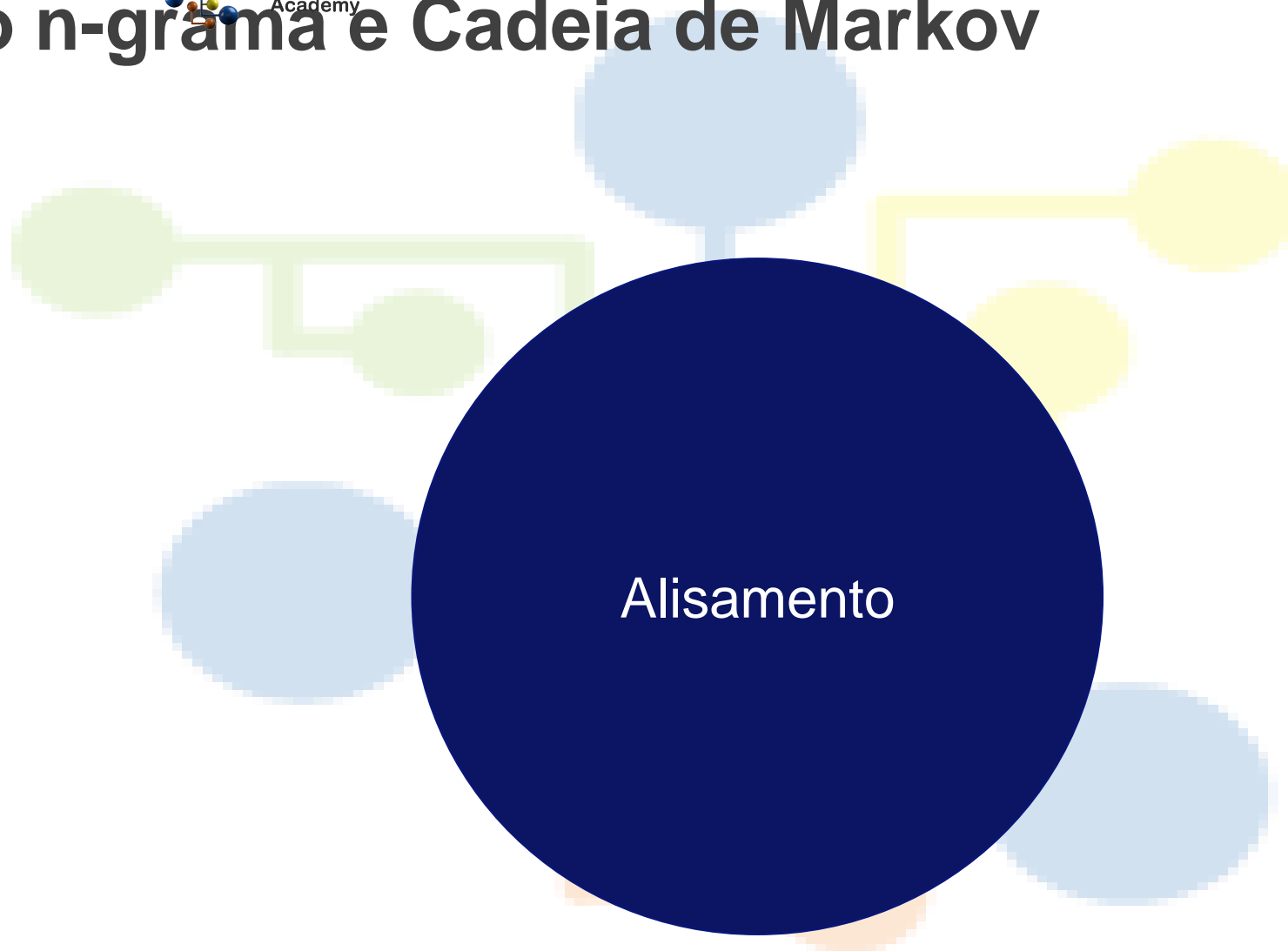
Isso significa que se deve atribuir  $P(\text{“ht”}) = 0$ ? Se fizermos isso, o texto “**The program issues an http request**” teria uma probabilidade de inglês zero, o que estaria errado!







# Modelo n-grama e Cadeia de Markov





# Modelo n-grama e Cadeia de Markov

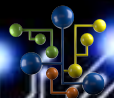
## Alisamento

O tipo mais simples de alisamento foi sugerido por Pierre-Simon Laplace, no século XVIII: ele disse que, na falta de mais informações, se uma variável aleatória booleana  $X$  for falsa em todas as  $n$  observações até agora, a estimativa de  $P(X = \text{verdadeiro})$  deve ser  $1/(n+2)$ .

$$\hat{P}(c_i | c_{i-2:i-1}) = \lambda_3 P(c_i | c_{i-2:i-1}) + \lambda_2 P(c_i | c_{i-1}) + \lambda_1 P(c_i)$$



# Avaliação de Modelo



# Avaliação de Modelo

A avaliação pode ser uma métrica de tarefa específica, tal como medir a precisão da identificação da linguagem.

$$\text{Perplexidade}(c_{1:N}) = P(c_{1:N})^{-\frac{1}{N}}$$





# Avaliação de Modelo

A perplexidade pode ser imaginada como o inverso da probabilidade, normalizada pelo comprimento da sequência.

$$\text{Perplexidade}(c_{1:N}) = P(c_{1:N})^{-\frac{1}{N}}$$



# Modelos de Palavras





# Modelos de Palavras

Agora voltamos aos modelos de  $n$ -grama de palavras em vez de caracteres.

Todos os mecanismos aplicam-se igualmente aos modelos de palavra e de caracteres.

A principal diferença é que o **vocabulário** — o conjunto de símbolos que compõem o *corpus* e o modelo — é maior.







# Modelos de Palavras

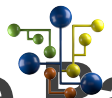
Quantas palavras existem em:

“ne'er-do-well”?

Ou

“Tel:1-800-960-5660x123”?





# Modelos de Palavras

Os modelos de palavras de  $n$ -grama têm de lidar com palavras **fora do vocabulário**.

Isso pode ser feito acrescentando apenas uma nova palavra ao vocabulário:  
<UNK>, como palavra desconhecida!

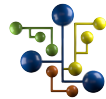


# Modelos de Palavras

Para se ter uma noção de que modelos de palavras criar, poderíamos construir modelos de unigrama, bigrama e trigrama usando as palavras de todos os capítulos do curso até aqui e, em seguida, sequências de palavras aleatoriamente amostradas dos modelos. Os resultados seriam:

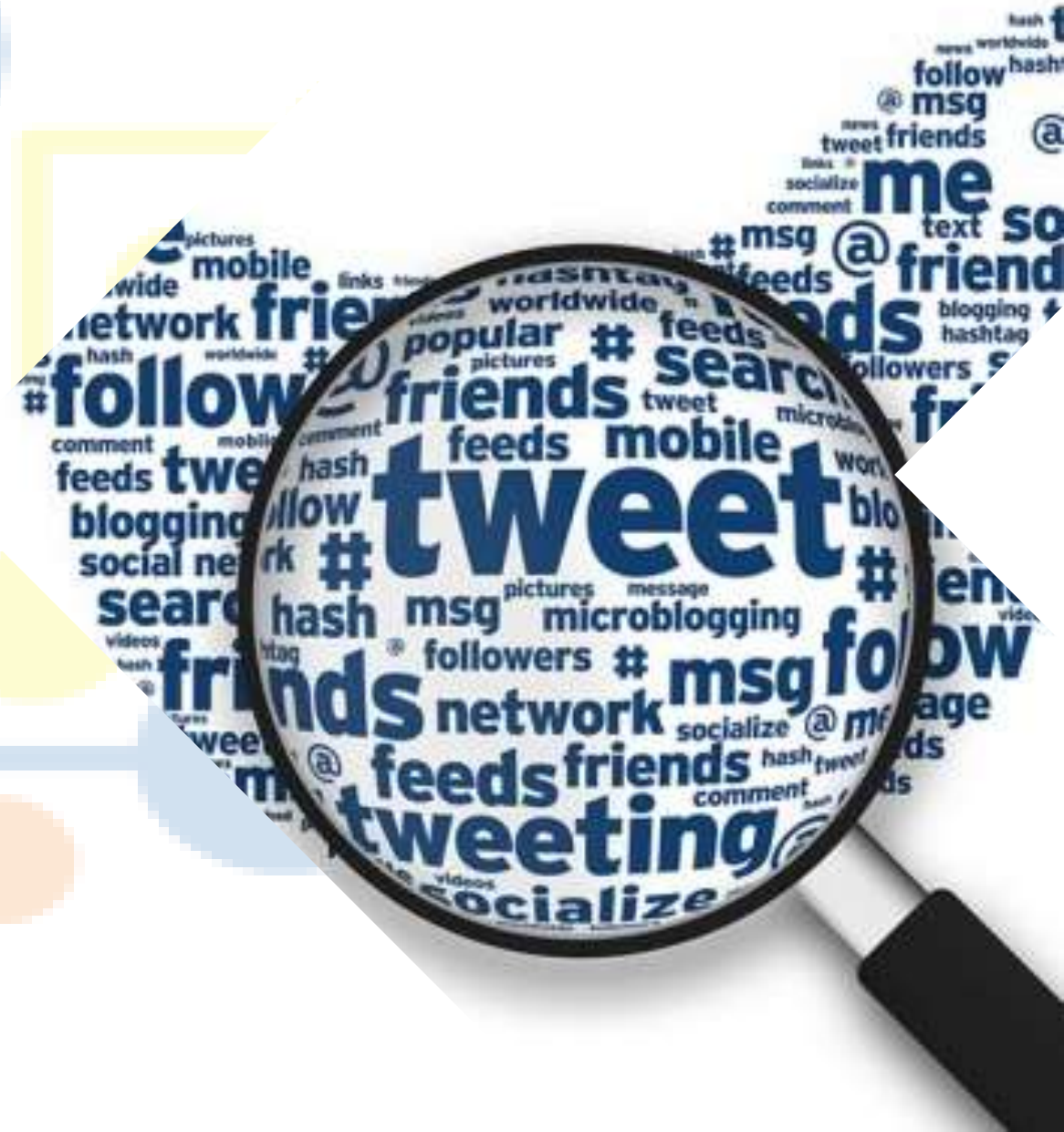
- **Unigrama:** lógicas são como são confusão um pode direito tentar agente objetivo foi...
- **Bigrama:** sistemas são abordagens computacionais muito semelhantes seria representado...
- **Trigrama:** planejamento e esquadulagem estão integrados o sucesso do modelo simples de bayes...

# Classificação de Texto

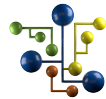


# Classificação de Texto

Vamos agora considerar em profundidade a tarefa de **classificação de texto**, também conhecida como **categorização**.







# Classificação de Texto

Uma vez que “não spam” é estranho,  
os pesquisadores cunharam o termo  
**ham** para “não spam”.





# Classificação de Texto

Spam: Venda por atacado de óculos modernos –57% hoje. Relógios de grife por preço baixo...

Spam: Você pode comprar ViagraFr \$ 1,85 Todos os medicamentos a preços imbatíveis!...

Spam: PODEMOS TRATAR DE QUALQUER COISA QUE VOCÊ TENHA APENAS CONFIE EM NÓS ...

Spam: Come.ce a ganhar\* o salário que vo,cê m-erece o'btendo referên'cias apropriada,s!

Ham: O significado prático de largura da hiperárvore identificando mais...

Ham: Resumo: Vamos motivar o problema de aglomeração de identidade social:...

Ham: É bom ver você meu amigo. Ei Pedro, Foi bom saber de você. ...

Ham: PDS implica convexidade do problema de otimização resultante (Kernel Ridge ...







# Classificação de Texto

Na abordagem de modelagem de linguagem, definimos um modelo de linguagem de  $n$ -grama para  $P(\text{Mensagem} \mid \text{spam})$  pelo treinamento em uma pasta de spam, e um modelo  $P(\text{Mensagem} \mid \text{ham})$  pelo treinamento em uma caixa de entrada. Então, podemos classificar uma nova mensagem com a aplicação da regra de Bayes:

$$\operatorname{argmax}_{c \in \{\text{spam}, \text{ham}\}} P(c \mid \text{mensagem}) = \operatorname{argmax}_{c \in \{\text{spam}, \text{ham}\}} P(\text{mensagem} \mid c) P(c)$$





# Classificação de Texto

A Regra de Bayes, novamente???





# Classificação de Texto

Na abordagem de modelagem de linguagem, definimos um modelo de linguagem de  $n$ -grama para  $P(\text{Mensagem} \mid \text{spam})$  pelo treinamento em uma pasta de spam, e um modelo  $P(\text{Mensagem} \mid \text{ham})$  pelo treinamento em uma caixa de entrada. Então, podemos classificar uma nova mensagem com a aplicação da regra de Bayes:

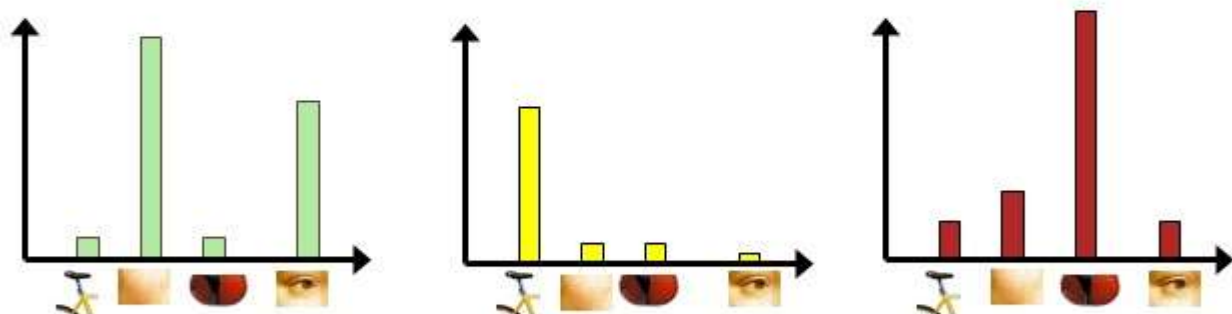
$$\operatorname{argmax}_{c \in \{\text{spam}, \text{ham}\}} P(c \mid \text{mensagem}) = \operatorname{argmax}_{c \in \{\text{spam}, \text{ham}\}} P(\text{mensagem} \mid c) P(c)$$





# Classificação de Texto

Na abordagem de aprendizagem de máquina representamos a mensagem como um conjunto de pares de característica/valor e aplicamos um algoritmo de classificação  $h$  à característica de vetor  $X$ .





# Classificação de Texto

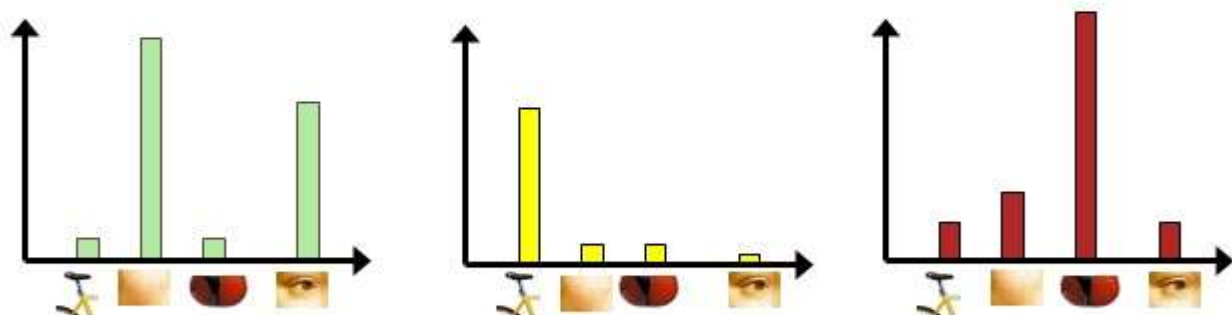
Essa representação do unigrama tem sido chamada de modelo de **saco de palavras** (bag of words).





# Classificação de Texto

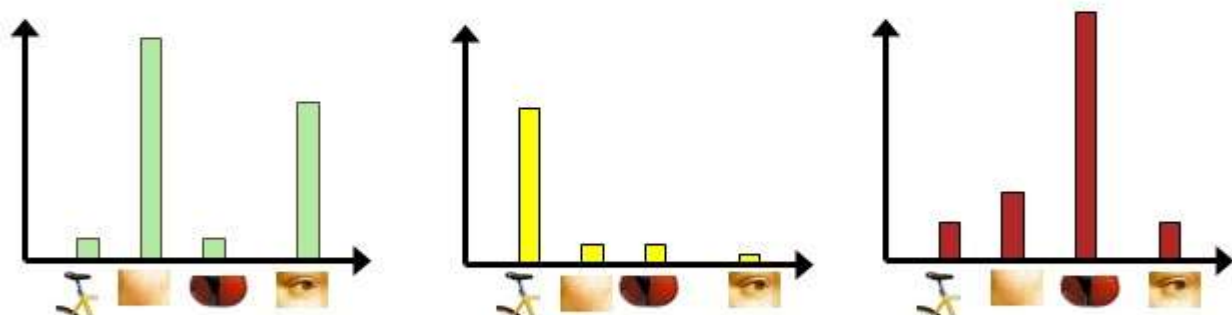
Com bigramas e trigramas, o número de características é elevado ao quadrado ou ao cubo e podemos acrescentar outras características não  $n$ -grama.





# Classificação de Texto

Pode ser caro executar algoritmos em um vetor de características muito grande, por isso muitas vezes um processo de **seleção de características (Feature Selection)** é usado para manter apenas as características que melhor distinguem entre spam e ham.







# Classificação de Texto

Uma vez escolhido um conjunto de características, podemos aplicar qualquer uma das técnicas de aprendizagem supervisionada que vimos; as mais populares para a categorização de texto incluem *k*-vizinhos mais próximos, máquinas de vetores de suporte, árvores de decisão, Naive Bayes e regressão logística.

# Recuperação de Informação



# Recuperação de Informação

A **recuperação de informação** é a tarefa de encontrar documentos que são relevantes para a necessidade de um usuário de obter informação.





# Recuperação de Informação

1. Um **corpus de documentos**. Cada sistema deve decidir o que quer tratar como documento: um parágrafo, uma página ou um texto de várias páginas.
2. **Consultas colocadas em linguagem de consulta**. Uma consulta específica sobre o que o usuário quer saber. A linguagem de consulta pode ser apenas uma lista de palavras, tais como [Curso IA] ou pode especificar uma sintagma com palavras que devem ser adjacentes, como em ["Curso IA"] e conter operadores booleanos como em [IA **E** Curso], incluir operadores não booleanos, tais como [IA PERTO curso] ou [Curso de IA site: [www.datascienceacademy.com.br](http://www.datascienceacademy.com.br)].
3. **Um conjunto de resultados**. Esse é o subconjunto de documentos que o sistema de RI julga ser **relevante** para a consulta. Por *relevante* queremos dizer provável que seja de utilidade para a pessoa que fez a consulta, para a necessidade de informação específica expressa na consulta.
4. **Apresentação do conjunto de resultados**. Isso pode ser tão simples como uma lista ordenada de títulos de documentos ou tão complexo como um mapa de cores de rotação do conjunto de resultados projetado em um espaço tridimensional, processado como uma exibição bidimensional.







# Recuperação de Informação

Os primeiros sistemas de RI trabalhavam com um **modelo de palavra-chave booleano**. Cada palavra na coleção do documento era tratada como uma característica booleana que era verdadeira em um documento se a palavra ocorresse no documento e falsa se não ocorresse.





# Recuperação de Informação

[agricultura (Paraná OU Santa Catarina)]

[informação E recuperação E modelos E otimização]

[informação OU recuperação OU modelos OU otimização]





# Recuperação de Informação

## Função de pontuação BM25

*Success*





# Recuperação de Informação

## **Função de pontuação BM25**

Na função BM25, a pontuação é uma combinação linear ponderada das pontuações para cada uma das palavras que compõem a consulta.



# Recuperação de Informação

## Função de pontuação BM25

Frequência  
do Termo

Frequência Inversa  
do Termo

Comprimento do  
Documento



# Recuperação de Informação

$$BM25(d_j, q_{1:N}) = \sum_{i=1}^N IDF(q_i) \cdot \frac{TF(q_i, d_j) \cdot (k + 1)}{TF(q_i, d_j) + k \cdot (1 - b + b \cdot \frac{|d_j|}{L})}$$

Frequência  
do Termo

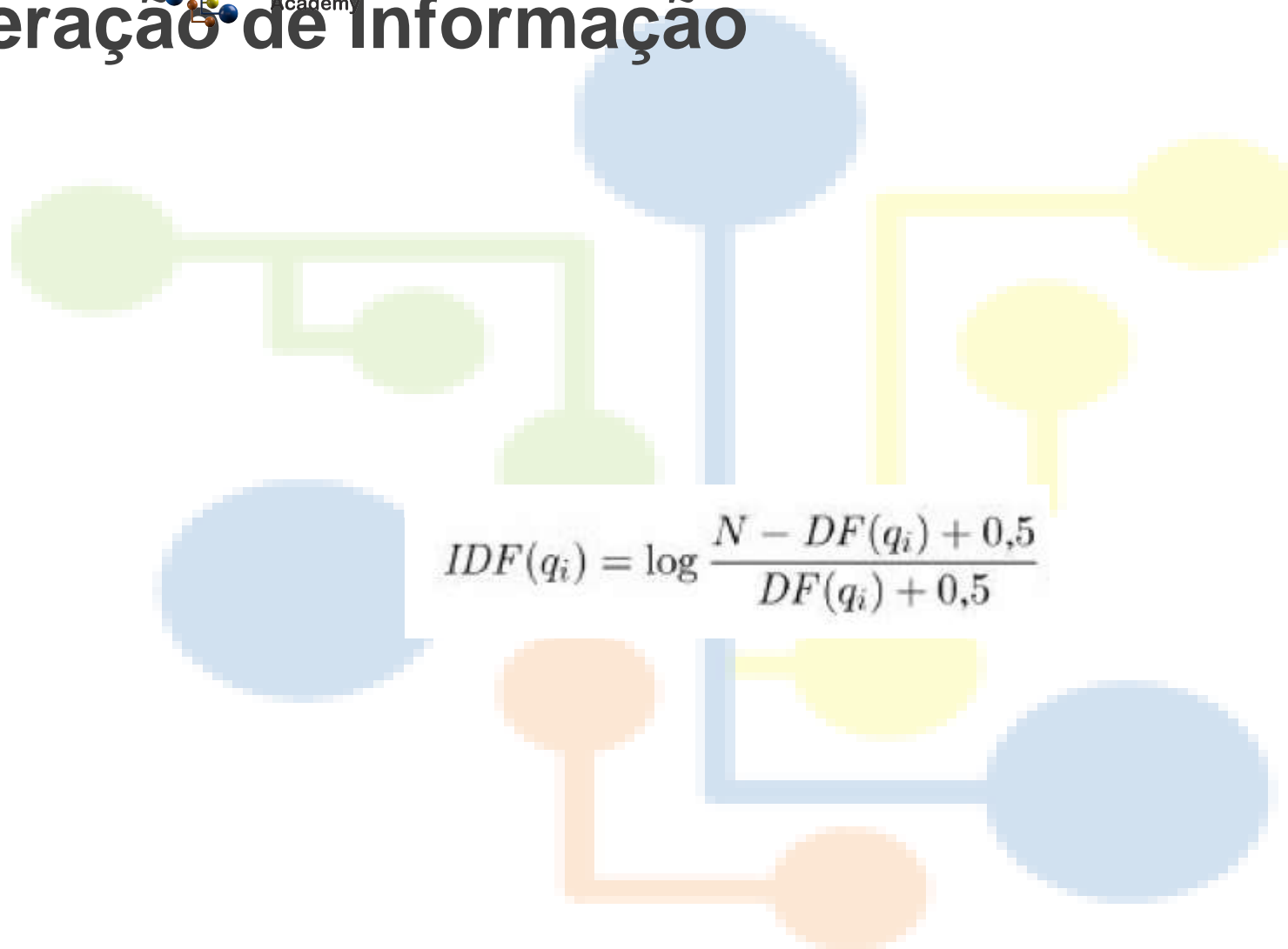
Frequência Inversa  
do Termo

Comprimento do  
Documento

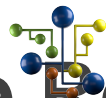




# Recuperação de Informação



# Avaliação e Refinamento dos Sistemas de RI



# Avaliação e Refinamento dos Sistemas de RI

	No conjunto de resultados	Fora do conjunto de resultados
Relevantes	30	20
Não relevantes	10	40

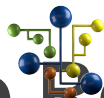
No nosso exemplo, a **precisão** é de  $30/(30 + 10) = 0,75$

A taxa de falsos positivos é de  $1 - 0,75 = 0,25$

No nosso exemplo, a **cobertura** é  $30/(30 + 20) = 0,60$

A taxa de falso negativo é de  $1 - 0,60 = 0,40$





# Avaliação e Refinamento dos Sistemas de RI

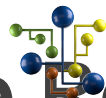
	No conjunto de resultados	Fora do conjunto de resultados
Relevantes	30	20
Não relevantes	10	40

É possível compensar a precisão em relação à cobertura variando o tamanho do conjunto de resultados retornado

Um resumo de ambas as medidas é a **pontuação de F1**, um número único que é a média harmônica de precisão e cobertura,  $2PR / (P + R)$ .



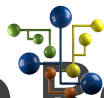




# Avaliação e Refinamento dos Sistemas de RI

Um refinamento comum é um modelo melhor do efeito do tamanho do documento sobre a relevância

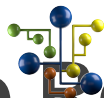




# Avaliação e Refinamento dos Sistemas de RI



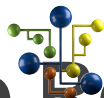
Por exemplo, se a consulta for [sofá], será uma pena excluir do conjunto de resultados aqueles documentos que mencionam “SOFÁ” ou “sofás” mas não “sofá”.



# Avaliação e Refinamento dos Sistemas de RI



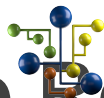
Por exemplo, transformar para o radical em inglês “**stocking**” para “**stock**” tenderá a diminuir a precisão para consultas sobre instrumentos financeiros, embora pudesse melhorar a cobertura para consultas sobre armazenagem.



# Avaliação e Refinamento dos Sistemas de RI



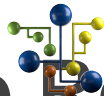
O próximo passo é reconhecer **sinônimos**, como “divã”, para “sofá”.



# Avaliação e Refinamento dos Sistemas de RI



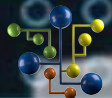
Um usuário que fornece a consulta  
[Tim Couch] quer ver resultados  
sobre o jogador de futebol e não  
sobre sofás.



# Avaliação e Refinamento dos Sistemas de RI

Como refinamento final, a RI pode ser melhorada considerando os **metadados** — dados externos ao texto do documento. Os exemplos incluem palavras-chave fornecidas por seres humanos e dados de publicação. Na Web, os **links** de hipertexto entre documentos são fonte crucial de informação.





Data Science  
Academy

Data Science Academy [raphaelbsfontenelle@gmail.com](mailto:raphaelbsfontenelle@gmail.com) 615c1fdde32fc361b30c9ec2

# Algoritmo PageRank



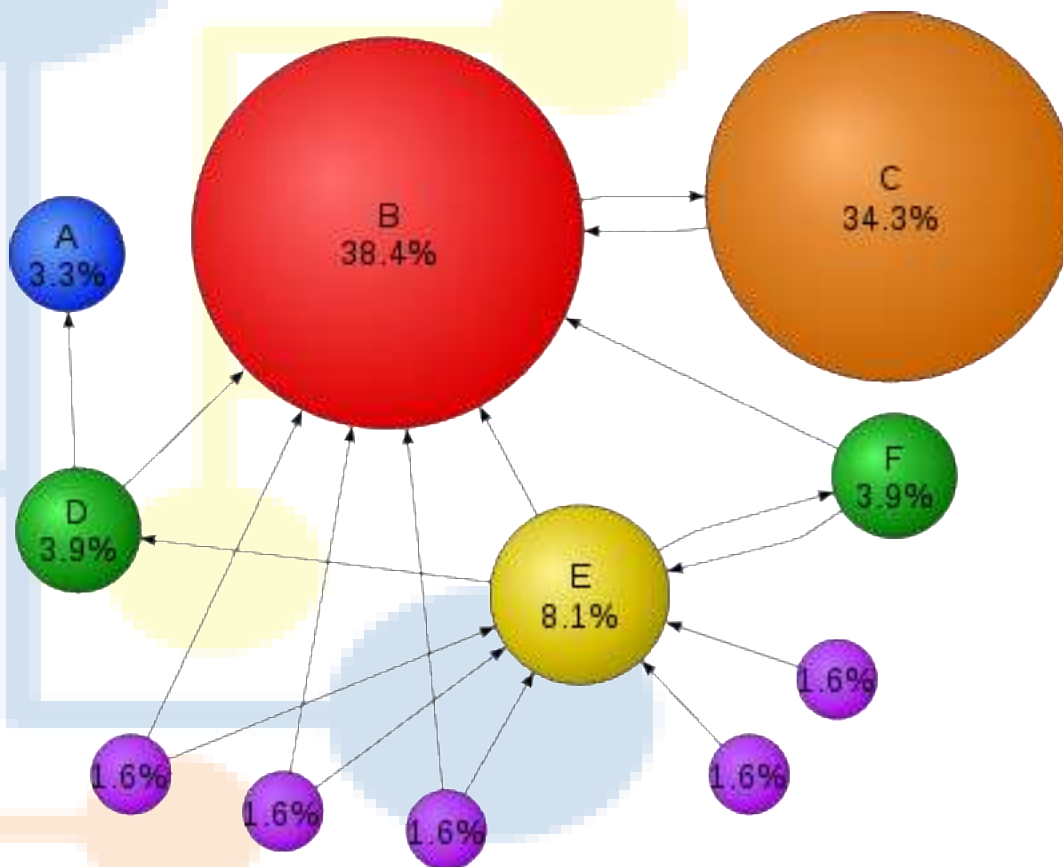
Data Science Academy





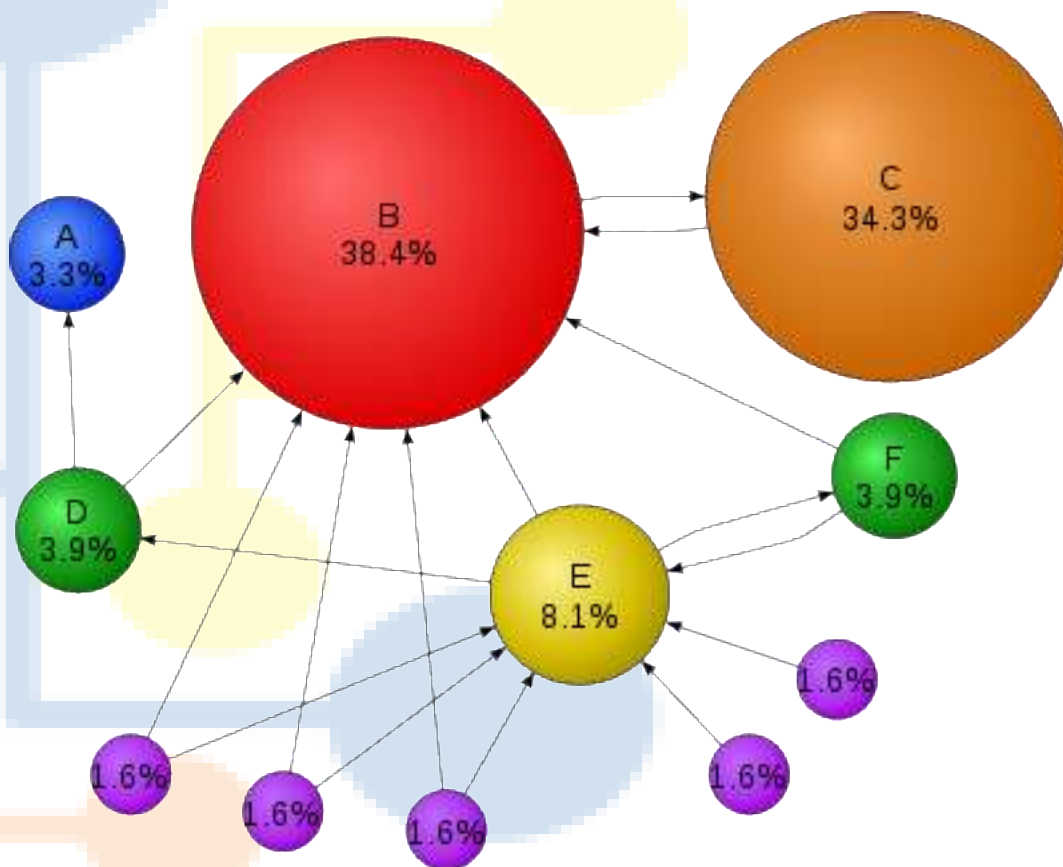
# Algoritmo PageRank

Se a consulta for [IBM], como podemos ter certeza de que a homepage da IBM, **ibm.com**, é o primeiro resultado, mesmo se outra página menciona o termo “IBM” com mais frequência?



# Algoritmo PageRank

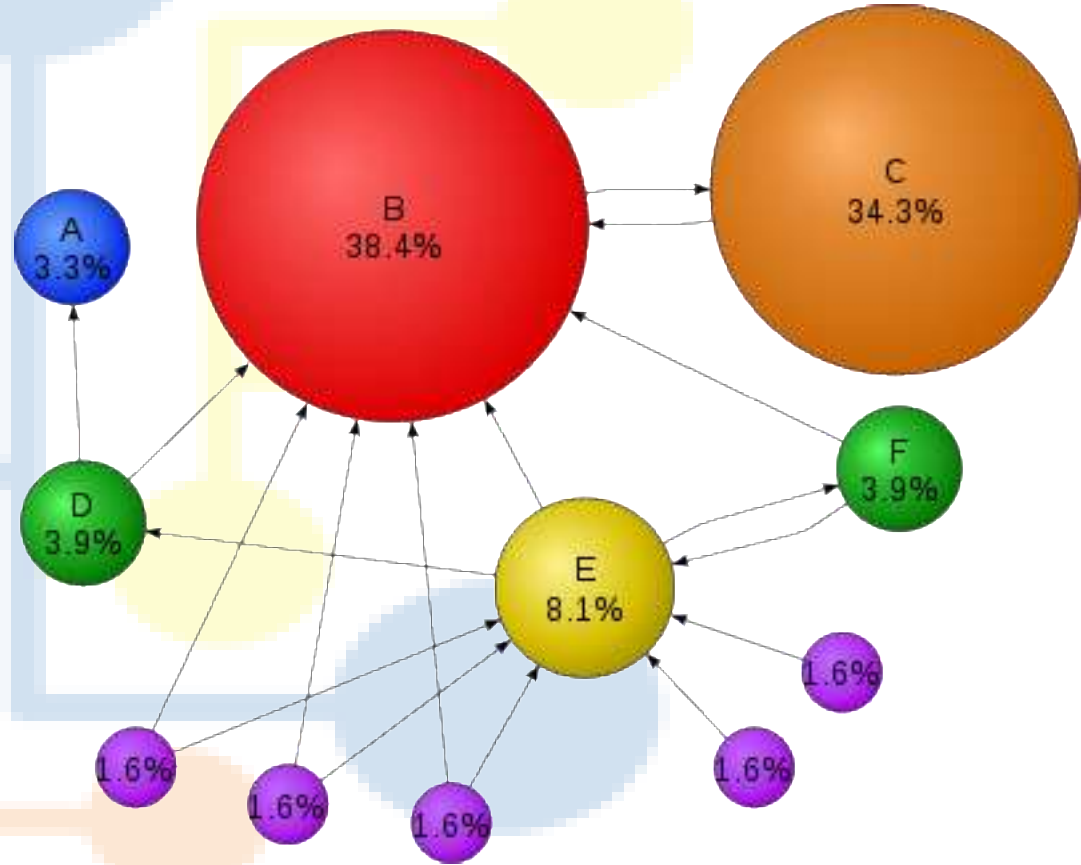
A ideia é que [ibm.com](http://ibm.com) tem muitas ligações de entrada (links para a página), por isso deve ter uma classificação mais alta: cada link de entrada é um voto para a qualidade da página vinculada.

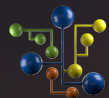




# Algoritmo PageRank

$$PR(p) = \frac{1-d}{N} + d \sum_i \frac{PR(in_i)}{C(in_i)}$$





Data Science  
Academy

Data Science Academy [raphaelbsfontenelle@gmail.com](mailto:raphaelbsfontenelle@gmail.com) 615c1fdde32fc361b30c9ec2

# Obrigado



Data Science Academy



Data Science Academy