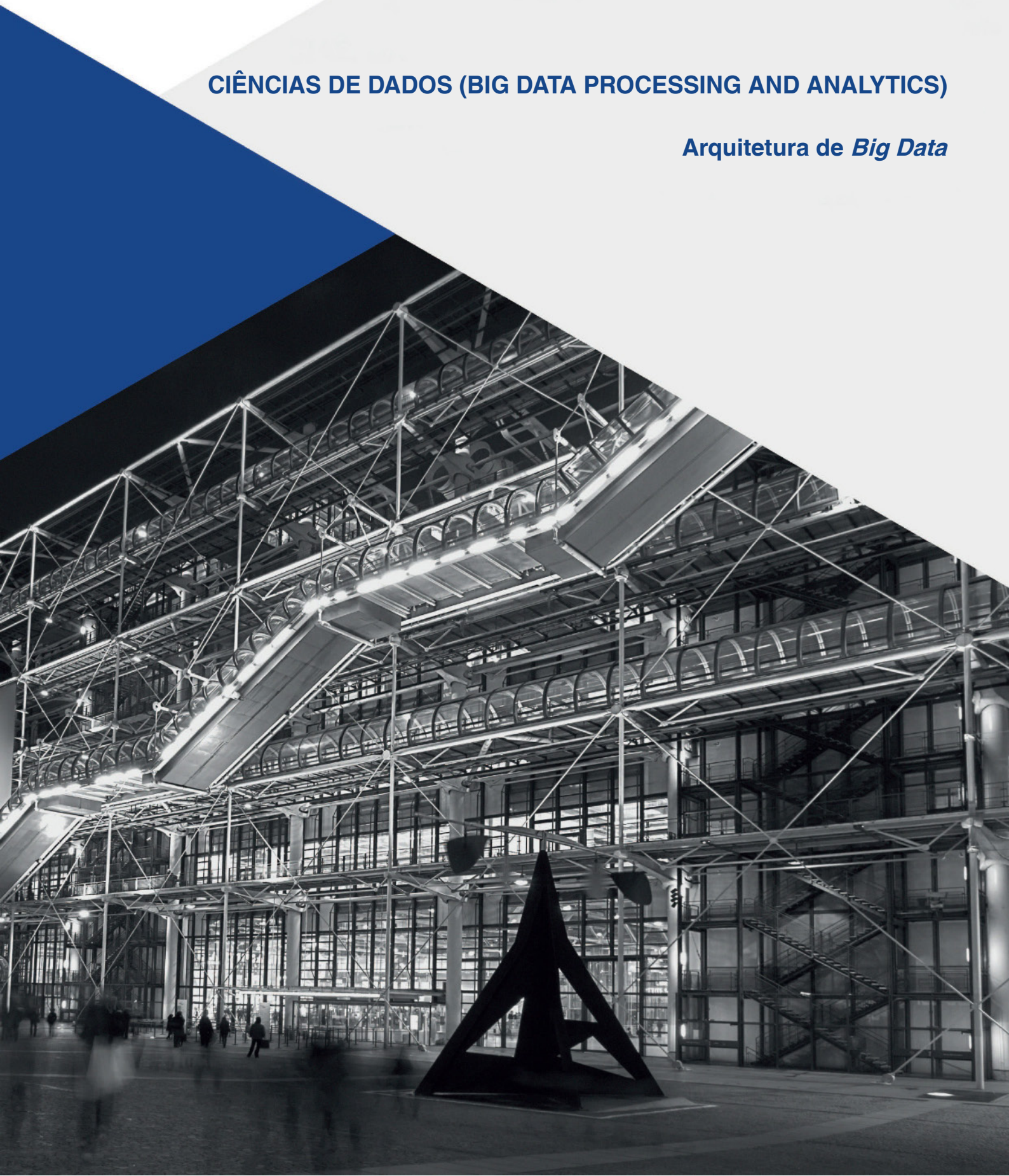
Universidade Presbiteriana Mackenzie

CIÊNCIAS DE DADOS (BIG DATA PROCESSING AND ANALYTICS)

Arquitetura de *Big Data*



Universidade Presbiteriana Mackenzie

Modalidade a distância

RECUPERAÇÃO DA INFORMAÇÃO NA WEB E EM REDES SOCIAIS

Trilha 1 — Introdução à Recuperação da Informação

Professor: Luciano Moreira Camilo e Silva

Sumário

1. Apresentação do componente curricular

O engenheiro e economista alemão Klaus Martin Schwab, fundador e presidente do fórum econômico mundial, escreveu, em 2015, para a revista **Foreign Affairs** que estamos vivendo a quarta revolução industrial..

A primeira revolução industrial ocorreu no início do século XIX e adveio das máquinas a vapor e da mecanização das linhas de produção; concentrou-se no Reino Unido.

A segunda revolução industrial ocorreu no final do século XIX e início do século XX, espalhando-se pela Europa, pelo Japão e pelos EUA. O conhecimento do eletromagnetismo permitiu uma nova revolução nos maquinários, com destaque aos meios de comunicação com o invento do telegrafo e, posteriormente, dos telefones.

A terceira revolução industrial começou há pouco mais de meio século e é também conhecida como a revolução digital. O invento dos transistores e os circuitos integrados levaram a humanidade a evoluções exponencias na área da computação, iniciada de forma ainda incipiente, com Charles Babbage, John Von Neumann e, claro, Alan Turing e Ada Lovelace, sob conceitos criados por seus pares séculos antes.

Por fim, a quarta revolução industrial, citada por Klaus e originada da pesquisa do governo alemão sobre o futuro da indústria e influência da internet das coisas (IoT, no acrônimo em inglês) (KAGERMANN et al., 2014) se caracteriza pelo extenso uso de automação, com uma comunicação persistente entre máquinas, por uso de modelos matemáticos computacionais do mundo real.

O objetivo deste componente curricular é prover conhecimento suficiente para que você possa compreender como os sistemas digitais lidam com informações e como utilizá-las no dia a dia. Iniciando pelos sistemas de organização e recuperação de informação, principalmente na web – quase onipresente hoje em dia – até o uso de ferramentas e algoritmos para tratamento e compreensão de textos (Processamento de Linguagem Natural, PLN).

Adicionalmente, pretende-se municiar o leitor deste e-book com uma compreensão de como esses dados podem ser usados para alavancar a quarta revolução industrial.

O componente está organizado em oito trilhas. As trilhas foram pensadas para construir uma jornada contígua, de modo que aprendizados na trilha 1 serão essenciais para uma compreensão profunda das demais trilhas. Cada trilha terá exemplos práticos, em cinco

desses módulos apresentados pelo professor e três mais complexos serão exercícios de fixação que você deverá realizar sozinho.

Além do conteúdo teórico, as trilhas se utilizarão primariamente de bibliotecas Python (versão 3.7) para os exemplos práticos.

É importante que você saiba que o objetivo das trilhas é ensiná-lo a pensar e "aprender a aprender", não devendo serem interpretadas, sob nenhuma hipótese, como um documento definitivo. Trata-se de um guia que o ajudará a compreender assuntos mais complexos, por meio da busca contínua de conhecimento que deve fazer parte do aprendizado do educando.

1.1. Organização das trilhas

• Trilha 1 – Introdução à recuperação de informação

Nessa trilha, serão introduzidos o conceito de recuperação de informação e o modo como ele se diferencia da recuperação de dados estruturados.

• Trilha 2 – Recuperação de informação por raspagem: Introdução

Nessa trilha, serão apresentadas a estrutura de árvore de um documento HTML e a maneira de obter informações contidas nestes documentos por meio da biblioteca Beautiful Soup do Python.

• Trilha 3 – Recuperação de informação por raspagem: robôs

Algumas vezes quer se buscar informações em diversas páginas de um mesmo site (busca de artigos em um blog) ou ainda em diversos sites distintos (busca de preço de hotel ou passagem área, por exemplo). Nessa trilha, serão introduzidas a biblioteca Scrapy do Python e a forma como construir seu próprio sistema de busca (webspider, em inglês).

• Trilha 4 – PLN: extração de palavras-chave

Computadores são excelentes para processamento de números, mas incapazes de entender textos. Nessa trilha, serão introduzidos os conceitos básicos de processamento de linguagem natural (PLN), como extrair palavras para montar um

saco de palavras (bag of words, em inglês) e identificar as principais palavras de um grupo de documentos.

• Trilha 5 – PLN: identificação de padrões de texto em documentos

Prosseguindo com o tema de processamento de linguagem natural, nessa trilha, serão expostas outras técnicas de pré-processamentos de textos e o algoritmo de tf-idf.

• Trilha 6 – PLN: análise de sentimentos

Encerrando as trilhas de processamento de linguagem natural, será apresentado o clássico caso de uso de análise de sentimentos em texto, com reforço dos conceitos trabalhados nas trilhas anteriores. Serão introduzidos também o conceito de APIs e os formatos de dados alternativos ao HTML.

• Trilha 7 – Redes Complexas: introdução a grafos

Iniciando na jornada com um tema novo, essa trilha explanará sobre conceitos básicos de grafo e os tipos mais comuns de redes complexas.

• Trilha 8 – Redes Complexas: uso em redes sociais na prática

Encerrando o componente, serão expostas as principais métricas utilizadas na análise de redes complexas e a forma como utilizá-las em análises de redes sociais.

2. Introdução à recuperação de informação

A recuperação de informação é algo que está tão associado a nossa rotina e tão presente nas buscas da internet que é fácil esquecer que se trata de uma ciência com décadas de existências.

A disciplina da recuperação de informação é mais antiga que a própria internet e é classificado como uma especialidade de Biblioteconomia, na área de Ciência da Informação, na grande área das Ciências Sociais Aplicadas.

Figura 1 – Tabela de classificação da área de conhecimento do CNPQ

6.00.00.00-7 Ciências Sociais Aplicadas
6.07.00.00-9 Ciência da Informação
6.07.01.00-5 Teoria da Informação
6.07.01.01-3 Teoria Geral da Informação
6.07.01.02-1 Processos da Comunicação
6.07.01.03-0 Representação da Informação
6.07.02.00-1 Biblioteconomia
6.07.02.01-0 Teoria da Classificação 6.07.02.02-8
Métodos Quantitativos. Bibliometria
6.07.02.03-6 Técnicas de Recuperação de Informação
6.07.02.04-4 Processos de Disseminação da Informação
6.07.03.00-8 Arquivologia
6.07.03.01-6 Organização de Arquivos

No final da década de 1950 e durante toda a década de 1960, o mundo já experimentava uma explosão de informações, e o uso de ferramentas, além das de bibliotecários, na organização dos dados era necessária, como descreveu Mooers ([s.d.], p. xx, tradução e destaque nossos):

> *Uma empresa que possui uma máquina de recuperação de informação, em geral, terá, adicionalmente, uma coleção especial com um bibliotecário responsável. A máquina é um sistema geralmente separado da biblioteca, exceto possivelmente para o armazenamento dos documentos. Em particular, isso significa que os aspectos intelectuais de lidar com a recuperação desses documentos de alta utilidade não são colocados nas mãos de um bibliotecário. Essa situação não é acidental e há boas razões para isso.*

A organização e busca de livros por assuntos e palavras-chave funcionava satisfatoriamente naquele momento. Entretanto, a busca por informação em jornais e outros periódicos,

que possuíam muitas informações similares, mas que se diferenciavam entre si, não era eficiente, quando tratada pelas técnicas tradicionais de classificação estudadas em biblioteconomia.

O sistema de recuperação, como proposto por Pedro (aaaa), pode ser definido pelo cumprimento dessas três etapas:

> • *Representação das informações contidas nos documentos, usualmente através dos processos de indexação e descrição dos documentos;*
>
> • *Armazenamento e gestão física e/ou lógica desses documentos e de suas representações;*
>
> • *Recuperação das informações representadas e dos próprios documentos armazenados, de forma a satisfazer as necessidades de informação dos usuários. Para isso é necessário que haja uma interface na qual os usuários possam descrever suas necessidades e questões, e através da qual possam também examinar os documentos atinentes recuperados e/ou suas representações.*

Em outras palavras, um sistema de recuperação de informação (SRI) tem como objetivo encontrar documentos, geralmente não estruturados, a partir dos dados inseridos pelo usuário. Utilizando algum sistema de indexação, outros algoritmos acessórios e organização desses documentos, o SRI manipula a consulta do usuário para entregar os documentos[1] que correspondem à necessidade do usuário.

Nos sistemas de recuperação de dados, onde é comum a aplicação da linguagem SQL (*Structured Query Language*)[2], espera-se que o retorno de dados à consulta do usuário seja exato, sem espaço para erros.

Ao contrário, nos sistemas de recuperação de informação, o resultado esperado do sistema é maximizar a relevância dos documentos retornados para as palavras solicitadas na consulta do usuário, e não um documento específico, ainda que o usuário tenha em mente exatamente o que deseja ver.

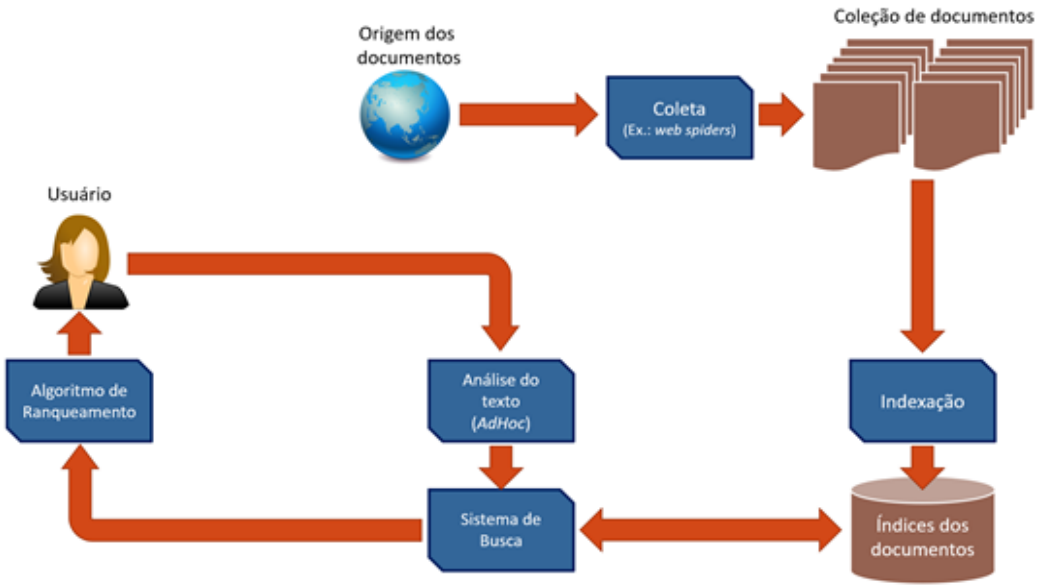
A arquitetura básica de um sistema de recuperação de informação, independente do perídio ou autor que a descreve, é comum e apresentada na Figura 2.

---

1 "Um documento é qualquer registro de informações, independentemente do formato ou suporte utilizado para registrá-las." ("Documento", Wikipédia).

2 "Structured Query Language, ou Linguagem de Consulta Estruturada ou SQL, é a linguagem de pesquisa declarativa padrão para banco de dados relacional (base de dados relacional). Muitas das características originais do SQL foram inspiradas na álgebra relacional." (SQL, Wikipédia).

Figura 2 – Diagrama da arquitetura básica de um sistema de recuperação de informação



2.1. Modelos de sistema de recuperação de informação

Como já dito anteriormente, sistemas de recuperação de informação existem há mais tempo que os sistemas de busca da web. Iniciaram utilizando máquinas mecânicas propostas por Mooers [s.d.],mas, inevitavelmente, esta disciplina encontrou apoio e sinergia na área da ciência da computação, especificamente nos sistemas de informação.

Muito evoluiu nos últimos 70 anos, e diversas variações de sistemas foram construídos. Foge do objetivo desse módulo, entretanto, discorrer sobre todas essas variações. Para aqueles mais interessados no assunto fica a recomendação do livro de Baeza-Yates e Ribeiro-Neto (2013), uma das referências no assunto, em texto didático e produzido em português.

Para o propósito dessa trilha, serão trabalhados determinados tipos de sistemas de recuperação da informação, com foco em sistema de recuperação de texto e sua evolução para recuperação de documentos de hipertexto.

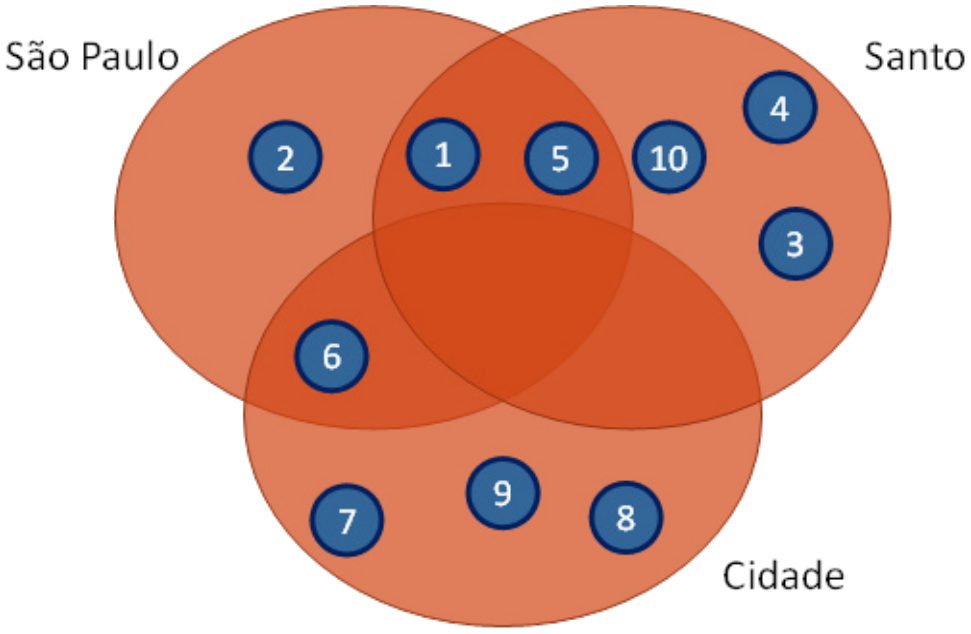
Acredita-se que os sistemas de recuperação na web se tornaram proeminentes no mundo atual, mas conhecer a história permitirá que você compreenda a jornada da evolução e, principalmente, aumente sua bagagem de conhecimentos.

Não se espera que você crie um sistema de recuperação de informação. Porém os princípios por detrás dos sistemas podem ser úteis em outros momentos de sua carreira.

O primeiro modelo, considerado o mais clássico, é o de busca booleana. Este modelo é apoiado na matemática da teoria dos conjuntos. Para cada documento, são extraídas todas as suas palavras, chamadas aqui de corpus, e executados alguns tratamentos, de modo a remover palavras menos relevantes para o contexto.

Não se preocupe com isso agora, as técnicas de tratamento serão mais bem exploradas nas próximas trilhas. Utilizando os operadores booleanos (veja o tópico 2.4.1 a seguir), o usuário pode auxiliar na redução do conjunto a ser retornado. No exemplo da Figura 3, ao informar que se busca por “São Paulo” e usando o operador booleano E (união) “cidade”, o SRI retornará apenas o documento 6.

Figura 3 – Representação visual de conjunto de documentos mapeados

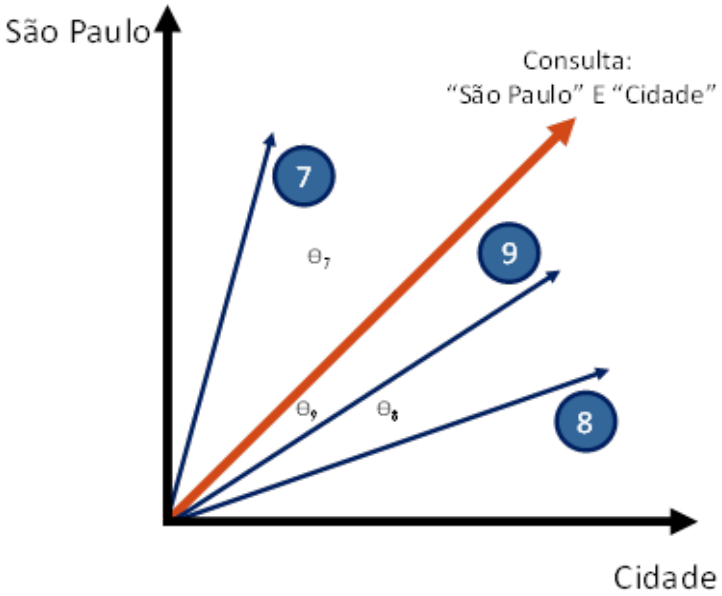


Esse modelo de busca de informação tem a vantagem de ser extremamente preciso, mas traz uma série de inconvenientes. O usuário precisa estar treinado para conseguir operá-lo com maestria e, o mais importante, tratar todos os documentos com o mesmo peso.

Uma proposta alternativa para o modelo booleano, proposta por Salton, Wong e Yang (1975) era o modelo vetorial. Cada palavra indexada pelo sistema de recuperação de informação tem um peso associado (falaremos mais de algumas técnicas nas trilhas futuras). Ao usuário pesquisar por um conjunto de palavras, um vetor multidimensional,

com um vértice em cada palavra desejada, era montado, e a ordem classificatória de retorno dos documentos era baseado nos ângulos entre os vetores, do menor (mais próximo da busca exata) ao maior.

Figura 4 – Representação gráfica da busca vetorial



Outra grande evolução veio de um artigo de Yu e Salton (1976), no qual propuseram uma evolução nos modelos. Tanto o modelo booleano clássico quanto os modelos vetoriais apresentavam um grave problema semântico: todas as palavras utilizadas na consulta tinham a mesma relevância na busca.

Com o modelo probabilístico, Yu e Salton (1976)encontraram uma forma de aumentar as chances de palavras mais relevantes para uma busca retornar documentos mais relevantes. O artigo traz uma matemática rebuscada e uma série de simplificações que não são verdadeiras para o mundo real, como assumir que a probabilidade de uma palavra aparecer no documento independe das outras palavras que existem no texto.

Como o próprio autor ressalta, entretanto, tais simplificações não se mostraram um problema para obter resultados de recuperação de informação mais relevantes. Principalmente para os sistemas de recuperação de informação que tinham um conjunto grande de documentos.

Diversas variações desses modelos, inclusive combinações entre si, surgiram ao longo dos anos. Contudo, dada a proeminência dos sistemas de recuperação de informações

na Web e o objetivo desse módulo em trabalhar com recuperação de informações da Web e Redes Sociais, o conhecimento histórico apresentado é suficiente.

A seguir, estudaremos dois tópicos importantes. Como utilizar índices para melhorar o desempenho das consultados aos dados (2.2) e o algoritmo de ranqueamento criado inicialmente pelo Google para ordenar os resultados (2.3)

2.2. Índices e índices invertidos

Na recuperação de dados, um dos instrumentos mais eficientes para buscar uma informação é a utilização de ordenação na chave primária.[3]

Como exemplo, imagine uma agenda telefônica, com atributos como Nome (do contato), Operadora (do celular), Empresa (onde a pessoa trabalha) e Número (de telefone). Imagine que você deseja buscar o contato chamado Pafúncio.

Sem utilizar-se de nenhum conhecimento adicional, a forma inocente de encontrar esse contato é verificando registro a registro do seu banco de dados.

Esta ação pode não demorar no caso de poucos contatos na agenda. Mas imagine uma agenda de contatos com mais de 200 milhões de brasileiros cadastrados? Mesmo que a operações de leitura de cada registro ocorra em apenas um milissegundo, o tempo de processamento para ler todos os registros e encontrar o desejado é superior a dois dias.

Com a informação de que a coluna do nome está ordenada e de que o total de registros é de 200 milhões, há uma forma mais esperta de se fazer essa busca. Inicia-se exatamente na metade da lista, lendo a posição número 100 milhões. Suponha que o nome encontrado seja “Luciano”. A partir dessa informação, infere-se que Pafúncio estará em algum registro posterior e evitou-se ler 100 milhões de registros.

Repete-se esse exercício, lendo a nova posição medial, 150 milhões. Encontra-se por hipótese o nome “Rodolfo” e, novamente, descarta-se todos os registros da posição 150

3 “Chaves primárias (em inglês, Primary keys ou “PK”), sob o ponto de vista de um banco de dados relacional, referem-se aos conjuntos de um ou mais campos, cujos valores, considerando a combinação de valores em caso de mais de uma chave primária, nunca se repetem na mesma tabela e, desta forma, podem ser usadas como um índice de referência para criar relacionamentos com as demais tabelas do banco de dados (daí vem o nome banco de dados relacional). Portanto, uma chave primária nunca pode ter valor nulo, nem repetição.” (Chave primária, *Wikipédia*).

milhões a 200 milhões. Em apenas duas execuções de leituras, reduziu-se a busca para os registros entre a posição 100 milhões e 150 milhões, ou 75% dos registros.

Repete-se essa operação até encontrar o registro desejado. Em menos de 30 leituras, no pior cenário, e dispendendo de menos de um segundo, encontra-se o registro desejado. **Esta é a beleza da busca binária.**

E como trabalhar com documentos e textos? O artificio acima só foi possível porque a busca era pelo nome exato do contato. Quando se busca um texto, insere-se apenas trechos ou palavras-chave, muitas vezes não contíguas da informação desejada.

Para exemplificar, pegue o poema de Carlos Drummond de Andrade (1928) como exemplo:

> *No meio do caminho tinha uma pedra*
> *Tinha uma pedra no meio do caminho*
> *Tinha uma pedra*
> *No meio do caminho tinha uma pedra*
> *Nunca me esquecerei desse acontecimento*
> *Na vida de minhas retinas tão fatigadas*
> *Nunca me esquecerei que no meio do caminho*
> *Tinha uma pedra*
> *Tinha uma pedra no meio do caminho*
> *No meio do caminho tinha uma pedra.*

Apenas para um exercício lúdico, imagine que cada linha do poema supracitado fosse um documento por si só. O método tradicional de armazenamento e indexação seria o mostrado na tabela abaixo.

Tabela 1 – Armazenamento dos textos de documento em uma tabela de banco de dados

Doc_ID	Doc_Text
1	No meio do caminho tinha uma pedra
2	Tinha uma pedra no meio do caminho
3	Tinha uma pedra
4	No meio do caminho tinha uma pedra
5	Nunca me esquecerei desse acontecimento
6	Na vida de minhas retinas tão fatigadas
7	Nunca me esquecerei que no meio do caminho
8	Tinha uma pedra
9	Tinha uma pedra no meio do caminho

10	No meio do caminho tinha uma pedra.

Fonte: Elaborada pelo autor.

Para resolver esse problema, como é explicado em Zobel e Moffat (2006) , utilizamos índices invertidos. A primeira etapa consiste em pegar cada texto da coleção e tratá-los da mesma forma como descrito no item 2.1, sobre os sistemas clássicos de recuperação da informação.

Com o resultado do tratamento, cria-se uma tabela com as principais palavras encontradas no corpus. E, para cada uma das palavras, insere-se o ID do documento em que ela foi encontrada. Por ser o contrário de uma indexação normal de banco de dados, esse índice ficou conhecido como índice (de textos) invertidos.

Tabela 2 – Tabela com a construção do índice invertido para o corpus acima

Word	Doc_Ids
acontecimento	5
caminho	1,2,4,7,9,10
esquecerei	5,7
fatigados	6
meio	1,2,4,7,9,10
nunca	5,7
pedra	1,2,3,4,8,9,10
retinas	6
tinha	1,2,3,4,8,9,10
vida	6

Fonte: Elaborada pelo autor.

A partir dessa tabela, caso o usuário pesquise pela palavra “fatigados”, pode-se usar a mesma técnica anterior e, rapidamente, encontrar que deve ser retornado apenas o documento 6.

2.3. Classificação de documentos e o Google pageRank

Ao analisar a Tabela 2, podemos perceber que usar a palavra "pedra" ou a palavra "caminho" não ajudará muito a identificar qual documento retornar ao usuário.

Para restringir os documentos retornados, no modelo booleano, usamos os operadores que explicaremos na seção 2.5. Ao buscar os documentos que possuem a palavra "pedra", mas não possuem a palavra "caminho", retornamos apenas os documentos 3 e 8.

Essa forma de busca, como já mencionada anteriormente, exige do usuário grande conhecimento na manipulação das consultas. Além disso, mesmo os modelos vetoriais e probabilísticos careciam de informações relevantes que não estavam contidas dentro do documento em si. Eram estatísticas frias extraídos apenas do próprio documento, e não da qualidade ou apelo que estes ofereceriam aos usuários.

Com um sistema de ordenação diferente, o Google conquistou praticamente um monopólio do mercado dos sistemas de recuperação da informação de sites da web.

Lawrence Edward Page, cofundador do Google, enquanto estudava para sua dissertação de doutorado em Stanford, teve uma ideia para um algoritmo de classificação. Pensando como um pesquisador acadêmico, Page fez uma analogia à World Wide Web como um conjunto de documentos sendo citados por outros documentos, como descrito por Battelle (2012, tradução e destaque nosso) na revista *Wired*:

> *Os acadêmicos constroem seus artigos sobre uma base de citação cuidadosamente construída: cada artigo chega a uma conclusão citando artigos publicados anteriormente como pontos de prova que avançam o argumento do autor. Os artigos são julgados não apenas por seu pensamento original, mas também pelo número de artigos que citam, pelo número de artigos que posteriormente os citam de volta e pela importância percebida de cada citação.*

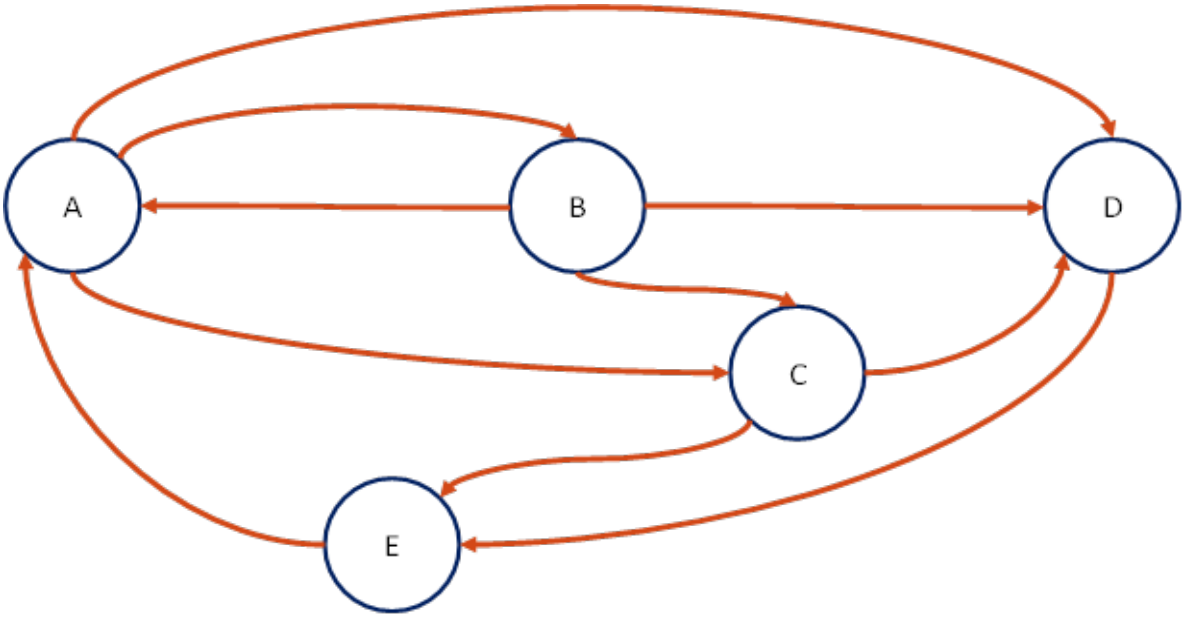
Com ajuda de outros alunos, foi publicado o artigo que deu origem ao Google, descrevendo o mecanismo base de classificação dos sites, nomeado pageRank[4] (PAGE et al., 1999).

Para exemplificar o exercício, imagine que tenhamos cinco sites, nomeados de A até E. As

4 Ao contrário do que muitos imaginam, o nome pageRank deriva no autor do artigo que deu origem ao algoritmo, Lawrence Edward Page. Sem dúvida, o trocadilho com a palavra inglesa page que também significa páginas (da internet) é conveniente e interessante.

setas indicam que um dos documentos referenciou o outro. Exemplo: A cita o documento D, e C cita o documento E. Como identificar qual é o documento mais relevante desse grafo?[5]

Figura 5 – Representação gráfica de cinco páginas com links entre si



Fonte: Elaborado pelo autor.

A primeira etapa do algoritmo do pageRank é calcular quantas citações cada documento tem, pois o peso da sua indicação será sua própria nota dividida pelo total de citações. Por exemplo, o documento A indica três outros documentos (B, C e D), logo, cada um deles recebera 1/3 dos pontos de A. O documento B indica apenas dois documentos (A e C), logo cada um deles receberá 1/2 dos pontos de B.

Inversamente, calcula-se os pontos de cada documento pela quantidade de indicações que ele recebeu. Por exemplo, o documento C recebe indicações dos documentos (A e B). Ou seja, a pontuação de C será igual a:

$$Pontos_C = Pontos_A \cdot {}^{1}/_{3} + Pontos_B \cdot {}^{1}/_{3}$$

5 Novamente, não se preocupe em entender grafos nesse momento. Será explanado melhor nas trilhas posteriores.

Ou de forma genérica:

$$Pontos_i = \sum_{j \neq i} \frac{Pontos_j}{\sum citações_j}$$

Inicia-se o processo assumindo que cada um dos documentos possui uma pontuação, por exemplo, 1,0 e calcula-se o valor resultante ao final da iteração. Repete-se essa iteração até o momento em que a menor diferença encontrada entre todos os documentos seja inferior a um estabelecido.

A Tabela 3 mostra o resultado de 20 iterações para o exemplo da Figura 3. Note que o total de pontos distribuído não se altera entre as iterações.

Tabela 3 – Cálculo iterativo do pageRank para o exemplo com cinco documentos

	Iteração 1	Iteração 2	Iteração 3	Iteração 4	Iteração 5	Iteração 10	Iteração 15	Iteração 20
A	1,00	1,33	1,61	1,65	1,35	1,53	1,50	1,50
B	1,00	0,33	0,44	0,54	0,55	0,49	0,50	0,50
C	1,00	0,67	0,56	0,69	0,73	0,65	0,67	0,67
D	1,00	1,17	0,89	0,96	1,07	0,99	1,00	1,00
E	1,00	1,50	1,50	1,17	1,31	1,35	1,33	1,33
Total	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00

Fonte: Elaborada pelo autor.

2.4. Operadores de busca em sistema de informação

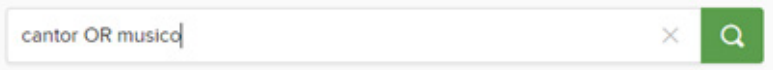
Os sistemas de recuperação de informação trabalham com processamento de linguagem natural. Dessa forma, para conseguir dar mais controle nos pedidos dos usuários, existem alguns operadores que podem ser utilizados em conjunto com a busca.

2.4.1. Operadores básicos

Operadores booleanos ou lógicos são aqueles que permitem você combinar palavras e indicar ao sistema de recuperação de informação como utilizá-las.

Esses comandos funcionam na maioria dos sistemas de recuperação de informação, sejam no seu site de busca favorito ou no sistema local da biblioteca da sua cidade.

- **Operador OR**: este operador ajuda você a buscar um documento quando não se tem clareza sobre os termos exatos que foram utilizados. Ele amplia a quantidade de resultados retornados pelo SRI:



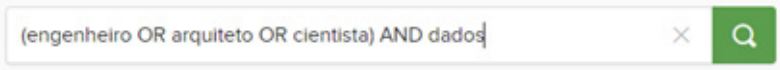
- **Operador AND:** funcionando de forma oposta ao operador OR, o operador AND restringe sua busca aos dois termos informados. Pouco usado, porque os sistemas de busca naturalmente adicionam o operador em uma busca. Procurar por engenheiro AND dados é equivalente a buscar engenheiro dados

- **Operador NOT (-):** acrescentar um sinal de menos antes de uma palavra informa ao sistema de recuperação de informação para excluir documentos que possuam aquela palavra. É útil para remover documentos com assuntos ambíguos, mas não desejados.

- **Aspas:** você se lembra que o operador AND é adicionado automaticamente pelos sistemas de recuperação de informação? Então! Para conseguir buscar uma frase ou palavra exata,[6] coloque-a entre aspas.

6 Atualmente, os sites de busca mais modernos procuram por sinônimos da palavra digitada. Use as aspas se você quer a correspondência exata.

• **Parênteses:** use o parêntesis para agrupar operações diferentes, como no exemplo a seguir, no qual queremos buscar um profissional de dados, seja arquiteto, cientista ou engenheiro.



2.4.2. Operadores avançadas

Com o tempo, os sistemas de recuperação da informação ficaram mais completos e complexos. E para atender à crescente demanda por busca de documentos, novos operadores foram criados.

Os operadores a seguir foram baseados no site de busca do Google, mas diversos desses operadores devem funcionar em outros sistemas modernos, principalmente os que buscam documentos na própria internet.

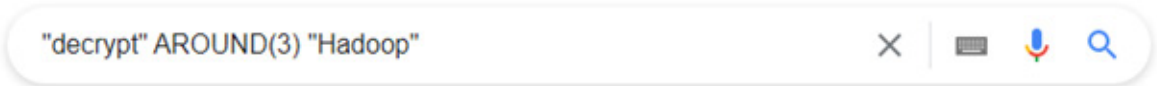
• **#..#:** Para buscar um documento que você não tem certeza dos números que apareceram, basta colocar o intervalo com dois pontos (..) entre o menor e o maior valor.

• **Intitle:** utilize a expressão intitle seguida de dois pontos (:) para garantir que a busca ocorrerá apenas nos títulos das páginas

• **inurl:** semelhante ao item anterior, porém, a busca se dará no endereço do site. Muito útil para buscar artigos publicados em algum ano específico, uma vez que é comum, nos blogs e sites de notícia, incluir a data da publicação na URL.

• **intext:** ao contrário do intitle, o intext ignora o título e busca apenas no conteúdo do site.

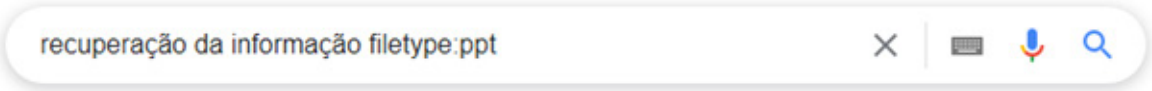
• **AROUND(X):** o mais incomum dos operadores. Utilizado entre duas palavras, ele faz a busca em qualquer documento que tenha até X palavras de distância entre as duas digitadas.



2.4.2. Operadores especiais do Google

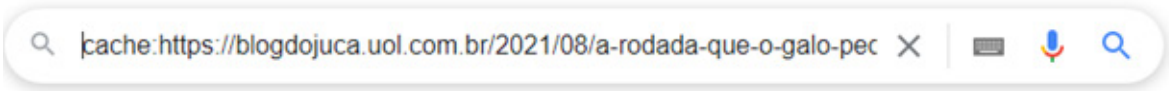
Para terminar essa lista de operadores especiais, existem alguns específicos do Google, mas que são extremamente úteis. Pode ser que funcionem em alguns de seus concorrentes.

• **Filetype:** operador para restringir os tipos de documentos buscados. Muito útil quando se quer buscar um PDF ou algum documento de apresentação do PowerPoint (PPT).



• **Site:** utilize esse operador caso você queira fazer uma busca apenas dentro de um site. Você pode restringir o domínio, digitando-o completamente (por exemplo: site:rodadas.anp.gov.br), ou ainda excluir somente um subdomínio como no exemplo abaixo.

• **Cache:** se um site estiver fora do ar, mas você quiser ver a cópia que o Google tem deste site, utilize esse operador. É necessário colocar o endereço completo do documento.



3. Síntese

Antes de seguirmos para processos de uso mais intenso dos dados e informações recuperadas na web e em redes sociais, é importante entendermos a história para compreender como chegamos aos níveis atuais de tecnologia.

Esta trilha tratou da evolução dos sistemas de recuperação da informação, desde os primórdios, quando eram utilizadas para pequenas coleções, com sistemas mecânicos, até os dias atuais e os sistemas de buscas como Google, Baidu e DuckDuckGo.

Ao longo da trilha, foram apresentados diversos conceitos criados ou aprimorados por essa ciência humana e com a ajuda intensa da subárea de sistemas da informação.

A extração de palavras-chave é um dos princípios por trás do processamento de linguagem natural, que será estudado mais a frente. O uso de índices invertidos é essencial para se trabalhar com textos, independentemente do sistema de ranqueamento.

O algoritmo do pageRank nos mostra como podemos usar redes complexas para obter o valor das relações entre diferentes documentos. Também vimos mais sobre redes complexas no final desta trilha.

Relembrando para alguns, descobrindo para tantos outros, terminamos esta trilha mostrando como tirar melhor proveito desses sistemas de recuperação da informação, principalmente na Web, foco deste componente. Alguém se lembra de usar os modelos booleanos nas bibliotecas há menos de duas décadas atrás?

Como exercício de aprendizado, utilize os operadores estudados nos sites de busca. Ganhe familiaridade com eles; isso, sem dúvida, irá auxiliá-lo em desafios que você não consegue imaginar agora. Por que não buscar artigos em PDF sobre sistema de recuperação da informação?

Para quem gosta de história e quer conhecer alguns sistemas alternativos de recuperação da informação aos apresentados aqui, pesquise sobre diretórios, como foram o Yahoo! e o finado Cadê! no Brasil.

E para aqueles que gostaram desta disciplina e querem se aprofundar nessa área das ciências sociais, saiba que há décadas de conteúdo produzido no mundo todo. Para encontrar, é só utilizar o sistema de recuperação da informação que melhor te agrada.

5. Referências

ANDRADE, C. D. No meio do caminho. *Revista de Antropofagia*, 1928, p. 1.

BAEZA-YATES, R.; RIBEIRO-NETO, B. *Recuperação de informação*: conceitos e tecnologia das máquinas de busca. 2. ed. Porto Alegre: Bookman, 2013.

BATTELLE, J. The Birth of Google. *Wired* (Condé Nast Digital), nov. 2012.

KAGERMANN, H.; WAHLSTER, W.; HELBIG, J. *Securing the future of German manufacturing industry:* recommendations for implementing the strategic initiative INDUSTRIE 4.0 Final report of the Industrie 4.0 Working Group. Relatório Final, Frankfurt, abr. 2013. Disponível em: <https://www.din.de/blob/76902/e8cac883f42bf28536e7e8165993f1fd/recommendations-for-implementing-industry-4-0-data.pdf>. Acesso em: 14 set. 2021.

MOOERS, C. N. Zatocoding and developments in information retrieval. *Aslib Proceedings*, v. 8, n. 1, p. 3-22, [s.d.].

PAGE, L.; BRIN, S.; MOTWANI, R.; WINOGRAD, T. The PageRank Citation Ranking: bringing order to the Web. *Technical Report*, São Francisco: Stanford InfoLab, 1999.

SALTON, G.; WONG, A.; YANG, C. S. A vector space model for automatic indexing. *Communications of the ACM*, v. 18, n. 11, 1975, p. 613-620.

SCHAWB, K. M. The Fourth Industrial Revolution. *Foreign Affairs*, 12 dez. 2015.

YU, C. T.; SALTON, G. Precision Weighting – an effective automatic indexing method. *Journal of the ACM,* v. 23, n. 1, 1976, p. 76-88.

ZOBEL, J.; MOFFAT, A. Inverted Files for text search engines. *ACM Computing Surveys*, v. 38, n. 2, 25 jul. 2006, artigo 6.