

Universidade Federal do ABC

**Summarticles: uma ferramenta para sumarização de artigos
científicos com inteligência artificial**

Projeto de Graduação em Computação

Aluno: Daniel Vieira Batista
Bacharelado em Ciência da Computação
daniel.batista@aluno.ufabc.edu.br

Orientador: Ronaldo Cristiano Prati
Centro de Matemática, Computação e Cognição
RonaldoPrati@ronaldo.prati@ufabc.edu.br

Santo André, 12 de dezembro de 2025

Daniel Vieira Batista

**Summarticles: uma ferramenta para sumarização de artigos
científicos com inteligência artificial**

Projeto de Graduação em Ciência da Computação
submetido ao Centro de Matemática, Computação
e Cognição da Universidade Federal do ABC,
para a obtenção do título de Bacharel em Ciência
da Computação.

Orientador: Prof. Ronaldo Cristiano Prati, Dr.

Santo André
2025

Resumo

Summarticles é uma ferramenta que extrai informações textuais de um conjunto de artigos científicos e sumariza a informação desses artigos utilizando visualizações analíticas, algoritmos de mineração de texto e algoritmos de inteligência artificial numa aplicação gratuita com interface Web. O intuito da ferramenta é facilitar o trabalho de cientistas na descoberta de conhecimento sobre um conjunto de artigos científicos. Dado o volume crescente de artigos científicos produzidos em diversos campos de pesquisa, torna-se difícil para os cientistas acompanharem toda a produção científica e priorizar quais artigos devem ser estudados, de acordo com suas necessidades de pesquisa. Neste sentido, Summarticles pode facilitar o trabalho de cientistas e pesquisadores.

Palavras-chave: inteligência artificial, sumarização de texto, processamento de linguagem natural, visualizações gráficas.

Abstract

Summarticles is a tool designed to extract textual information from a corpus of scientific articles and summarize their content using analytical visualizations, text mining, and artificial intelligence algorithms within a free, web-based interface. The tool's primary objective is to facilitate the knowledge discovery process for scientists regarding collections of scientific literature. Given the increasing volume of articles published across various research fields, it has become challenging for scientists to keep pace with scientific output and prioritize studies according to their specific research needs. In this context, Summarticles serves to streamline the work of scientists and researchers.

Keywords: artificial intelligence, text summarization, natural language processing, graphical visualizations.

Sumário

Lista de Tabelas

Lista de Figuras

1	Introdução	13
1.1	Motivação deste projeto	13
1.2	Objetivo geral	15
1.3	Objetivos específicos	15
2	Pesquisas Relacionadas	16
2.1	Bibliometria	16
2.2	Ferramentas semelhantes	17
3	Fundamentação Teórica	19
3.1	O Processo de Descoberta de Conhecimento	19
3.2	Tipos de Dados	19
3.2.1	Dados Estruturados	19
3.2.2	Dados Semi-Estruturados	20
3.2.3	Dados Não Estruturados	21
3.3	Artigo Científico	22
3.3.1	Estrutura de um Artigo Científico	24
3.4	Processamento de Linguagem Natural	26
3.4.1	Pré-processamento de Textos	28
3.5	Representação vetorial de dados textuais	35
3.5.1	Bag of Words (BOW)	36
3.5.2	TF-IDF	39
3.5.3	Vetores latentes (<i>embeddings</i>)	40
3.5.4	Similaridade de Textos	44
3.5.5	Agrupamento de Textos	45
3.5.6	Mineração de Tópicos	46
3.6	Inteligência Artificial	48
3.6.1	Breve História da Inteligência Artificial	49
3.6.2	Campos da Inteligência Artificial	51
3.7	Aprendizado de Máquina	52
3.7.1	Aprendizado Supervisionado	52
3.7.2	Redes Neurais Artificiais	53
3.7.3	Aprendizado Não-Supervisionado	55
3.7.4	Agrupamento	56
3.7.5	Redução de Dimensionalidade	58
3.7.6	Arquitetura <i>Transformers</i>	59
3.7.7	Grandes Modelos de Linguagem (ou <i>Large Language Models - LLMs</i>)	60

3.7.8	Geração Aumentada via Recuperação (RAG)	63
3.7.9	Outras Categorias de Aprendizado	64
3.8	Visualização de Dados	65
3.8.1	Sistema de Percepção Humana	66
3.8.2	Técnicas de visualização	67
3.8.3	Visualizações de Dados	69
4	Conjunto de dados utilizado	79
4.1	Dados Extraídos de Artigos	79
4.1.1	Metadados	80
5	Ferramenta Computacional Summarticles	81
5.1	Ambiente de Desenvolvimento	81
5.1.1	Dispositivo utilizado	82
5.1.2	Docker	82
5.1.3	GROBID	82
5.1.4	Ollama	83
5.1.5	Python	83
5.1.6	Streamlit	84
5.2	Diagramas da Aplicação Summarticles	84
5.2.1	Diagrama de Componentes	84
5.3	Utilizando a ferramenta Summarticles	86
5.3.1	Instalação da ferramenta Summarticles	86
5.3.2	Executando a Ferramenta Summarticles	87
5.3.3	Entrada da Ferramenta Summarticles	87
5.3.4	Saída da ferramenta Summarticles	89
6	Comparação de Ferramentas	90
7	Considerações finais	92
7.1	Limitações	93
7.2	Trabalhos futuros	94
Referências Bibliográficas		95

Lista de Tabelas

1	Exemplo de aplicação dos conceitos de processamento de linguagem natural. Utilizando uma passagem do livro De Assis (1998).	30
2	Tokenização de frase do livro Dom Casmurro de de Assis (1899).	30
3	Exemplo de normalização de frase do livro Dom Casmurro de de Assis (1899) .	30
4	Exemplo de aplicação dos conceitos de stemização e lematização.	30
5	Exemplo de aplicação do conceito de n-grams.	34
6	Documentos que serão utilizando no <i>Bag of Words</i>	38
7	Documentos normalizados para o <i>Bag of Words</i>	38
8	Vocabulário do <i>Bag of Words</i>	38
9	Matriz de frequência do <i>Bag of Words</i>	38
10	Matriz de frequência do <i>Bag of Words</i>	38
11	Matriz de frequência genérica do <i>Bag of Words</i>	38
12	Comparação entre ferramentas	90

Lista de Figuras

1	Produção anual de artigos científicos indexados na base de dados <i>SCImago</i> categorizado por continente (Oliveira <i>et al.</i> , 2022).	14
2	Um exemplo de visualização da rede de coautoria universitária, na ferramenta bibliométrica <i>VOSviewer</i> , construída por meio de contagem completa disponível em Perianes-Rodriguez <i>et al.</i> (2016). Uma visualização interativa está disponível em van Eck & Waltman (2010).	17
3	Um pequeno conjunto de dados estruturados clássico de Fisher (1936b). Um dos primeiros conjuntos de dados conhecidos usados para avaliar métodos de aprendizado de máquina na tarefa de classificação, disponível em Fisher (1936a).	20
4	Um exemplo de artigo pré-processado pela ferramenta Summarticles e transformado em um arquivo do tipo .xml semi-estruturado.	21
5	Um exemplo de dado não estruturado são vídeos e imagens. A imagem mostra uma playlist do YouTube que possui dados semi-estruturados e não estruturados, disponível em Stanford Online (2021).	21
6	Estrutura IMRD do artigo científico de acordo com Ribeiro (2016).	22
7	Exemplo de expressão regular usada na normalização de dados textuais em Summarticles.	31
8	Diagrama de frequência das palavras. Quanto mais frequente ou extrema é uma palavra, menor é sua relevância em termos de informação relevante para compreensão do dado textual (Catae, 2012).	33
9	A curva de Zipf e os cortes de Luhn adaptados de Matsubara <i>et al.</i> (2003). . .	34
10	Representação do espaço vetorial de documentos D_i em função de seus termos T_i , adaptado de (Salton <i>et al.</i> , 1975).	36

11	A árvore evolutiva dos <i>Large Language Models</i> (LLMs) modernos traça o desenvolvimento de modelos de linguagem, adaptado de Yang <i>et al.</i> (2024). A base destes modelos estão os modelos iniciais de embeddings, como por exemplo o Word2Vec (Mikolov, 2013).	41
12	As duas técnicas usadas por Word2Vec, adaptado de Mikolov (2013).	43
13	Representação gráfica da similaridade de cossenos.	45
14	Medidas de similaridade, adaptado de Gomaa <i>et al.</i> (2013).	46
15	Processo de agrupamento do dado textual por etapas.	46
16	Processo de geração de tópicos pelo LDA, adaptado de Mahmood (2009). . . .	48
17	Mapa da fronteira de conhecimento da inteligência artificial JSAI (2021). . . .	49
18	Diagrama de linha do tempo mostrando a história da inteligência artificial adaptado de Bellini <i>et al.</i> (2022).	50
19	Campos da IA segundo JSAI, 2021.	51
20	Um diagrama esquemático que descreve o modelo de aprendizado supervisionado, adaptado de Muhammad & Yan (2015).	53
21	Uma representação do neurônio artificial de McCulloch-Pitts, adaptado de Haykin (2001); McCulloch & Pitts (1943).	54
22	Uma representação da rede neural multi-perceptron de alimentação direta (<i>feed-forward</i>), adaptado de Popescu <i>et al.</i> (2009).	55
23	Uma representação em duas dimensões de uma agrupamento de dados utilizando o algoritmo <i>K-means</i> , adaptado de Burkardt (2009).	57
24	O processo de evolução das gerações de modelos de linguagem de acordo com Zhao <i>et al.</i> (2023).	61
25	O processo de evolução das gerações de modelos de linguagem de acordo com Zhao <i>et al.</i> (2023).	62
26	O processo de RAG aplicado num contexto de perguntas e respostas de acordo com Gao <i>et al.</i> (2023).	64

27	Comparação entre os três paradigmas do mecanismo RAG de acordo com Gao <i>et al.</i> (2023).	65
28	O processo biológico do sistema visual humano de acordo com Josko, 2023.	67
29	A imagem acima foi retirada do trabalho de Pernice <i>et al.</i> (2010) e mostra o mapa de calor formado pelas áreas mais vistas por usuários de um <i>website</i> , foi construído com dados de <i>eyetracking</i> . Observa-se o padrão F de leitura expresso pelos usuários no processamento da informação digital.	68
30	Imagen adaptada do livro de Tufte & Graves-Morris, p.51, na imagem o autor descreve os princípios dos gráficos de excelência para visualizações gráficas baseadas em dados informacionais.	70
31	Um exemplo de gráfico de linha, perceba que há anotações, cores e mais de uma linha no gráfico, combinando diferentes técnicas de visualização.	71
32	Um exemplo de gráfico de barra comparando dois grupos (cores) ao longo do tempo (eixo X) numa dimensão numérica (eixo Y).	72
33	Um exemplo de gráfico de pontos, onde observamos a combinação de cores, ícones e duas dimensões numéricas.	73
34	Um exemplo de gráfico de áreas, perceba a combinação de diferentes tipos de visualizações e técnicas.	74
35	Um exemplo de gráfico de setores	75
36	Um exemplo de gráfico matriz que representa um diagrama de mapa de calor, pois combina uma escala de cores, um gradiente que varia conforme a variação de uma variável numérica.	76
37	Um exemplo de grafo, traz uma relação hierárquica entre os pontos. Esse tipo de diagrama pode ser combinado com cores, tamanho e ícones para gerar visualizações complexas e com grande quantidade de informação.	77
38	Um exemplo de gráfico de nuvem de palavras (<i>Word Cloud</i>), adaptado de Hearst <i>et al.</i> (2019). Perceba que palavras maiores possuem valores maiores, combina-se também cores ou ícones na visualização de dados.	78

1 Introdução

Cientistas e pesquisadores costumam acompanhar seus campos de pesquisa por meio da leitura de periódicos, revistas científicas, participação em conferências, simpósios, seminários e cursos. A grande maioria das novidades científicas é divulgada por meio de artigos publicados em repositórios como *ResearchGate*, *arXiv*, *SciELO*, *PubMed*, *Google Scholar*, *JSTOR*, *DOAJ*, *Academia.edu*, entre muitos outros repositórios de conhecimento científico. A produção científica moderna tem como produto final um conteúdo principalmente textual, normalmente em formato digital.

Portanto, surge uma questão comum entre pesquisadores de várias disciplinas: como encontrar publicações relevantes para sua pesquisa, selecioná-las com base em critérios confiáveis, tratar os dados e aplicar as informações obtidas em seus projetos (de Medeiros *et al.*, 2015).

Em todo projeto científico, é essencial realizar uma pesquisa bibliográfica para a geração de novos conhecimentos científicos. Entre os objetivos das pesquisas bibliográficas, destaca-se a utilização de trabalhos desenvolvidos por outros pesquisadores como material de referência. No entanto, a grande quantidade de publicações nas diversas áreas de pesquisa apresenta desafios em relação à coleta e seleção cuidadosa.

1.1 Motivação deste projeto

Estima-se que mais de 50 milhões de artigos acadêmicos foram publicados até 2010 (Jinha, 2010). De acordo com Oliveira *et al.* (2022), de 1996 a 2018, o ritmo de crescimento da produção científica global foi, em média, de 7,4% ao ano, como mostra a Figura 1. Landhuis (2016) mostra que, nas últimas décadas, o número de artigos científicos publicados aumentou de 8% a 9% ao ano. Esse ritmo de crescimento corresponde ao dobro do período de 1981-1994, quando a produção mundial de artigos científicos aumentou 3,7% ao ano (Vargas, 1997).

De acordo com o Relatório intitulado ”*Science powers commerce—but not only*” (Dufour, 2015) publicado em 2015, entre os anos de 2008 e 2014, o número de artigos científicos catalogados no *Web of Science* (WoS) cresceu 23%. A produção científica vem crescendo ao longo das décadas em todos os continentes como mostra a Figura 1.

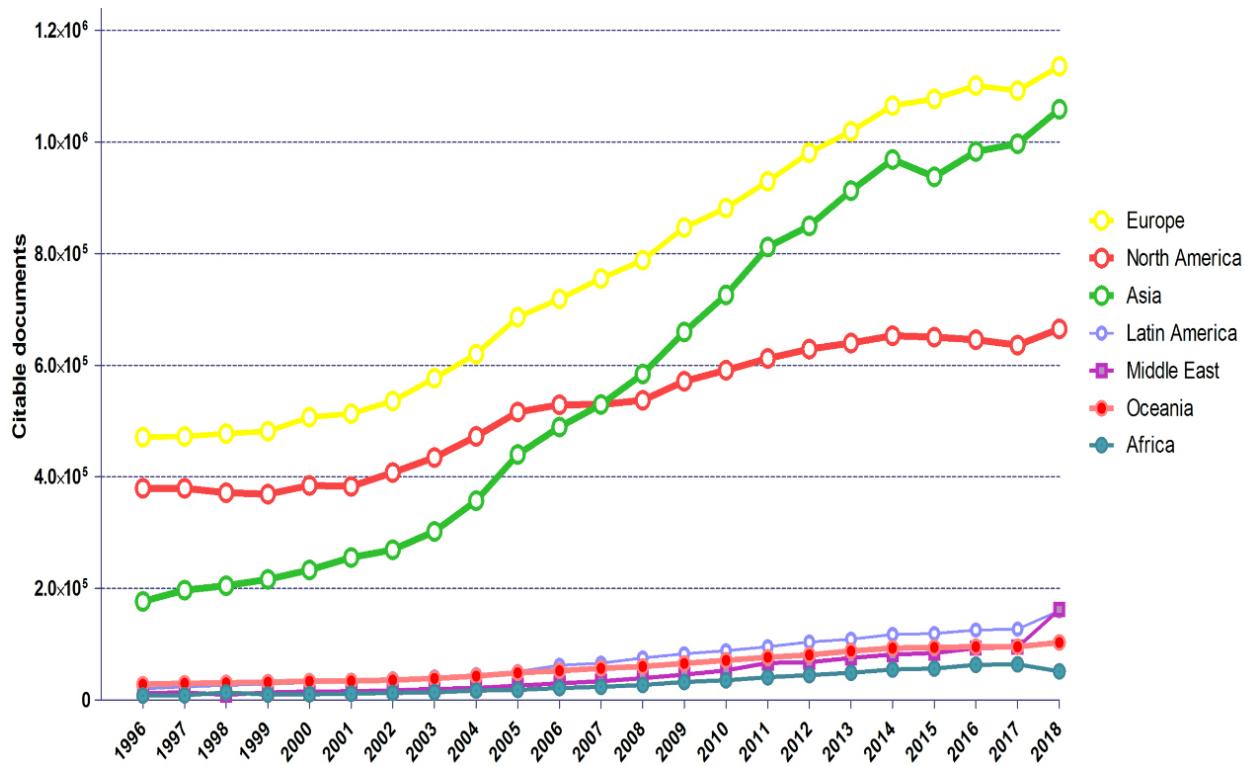


Figura 1: Produção anual de artigos científicos indexados na base de dados *SCImago* categorizado por continente (Oliveira *et al.*, 2022).

Nos últimos anos, o número de artigos publicados cresceu consideravelmente, segundo Bornmann & Mutz (2015). Em 2016, cerca de 1,92 milhão de artigos foram indexados pelos bancos de dados de publicações Scopus e *Web of Science*. Já em 2022, apenas seis anos depois, esse número cresceu mais de 45%, saltando para 2,82 milhões de artigos (Hanson *et al.*, 2023).

Esse crescimento da produção científica é muito importante para a evolução da ciência e da humanidade (Lichtfouse, 2013). Porém, impõe-se uma nova realidade para os pesquisadores, pois acompanhar a evolução de um campo científico ou manter-se atualizado sobre a produção científica torna-se morosa, dado o volume de informação gerado.

A busca por referências bibliográficas, uma etapa essencial do trabalho acadêmico, exige rigor na busca, seleção e análise de dados. Nesse processo, a revisão sistemática e a análise biométrica da literatura desempenham um papel importante, visando integrar protocolos quantitativos para a pesquisa de referências.

1.2 Objetivo geral

O objetivo geral deste Projeto de Graduação em Computação é o desenvolvimento de uma ferramenta de software livre para a extração e descoberta de informações de um conjunto de artigos científicos. Além disso, outro objetivo é exercitar as competências e habilidades desenvolvidas durante a minha formação acadêmica de graduação em Ciência da Computação.

1.3 Objetivos específicos

- Exercitar competências e habilidades da formação acadêmica de um cientista da computação;
- Aprender e utilizar algoritmos de inteligência artificial, com ênfase em processamento da linguagem natural;
- Desenvolver uma ferramenta de *software* que extrai informações de artigos científicos e as sumariza;
- Elaborar visualizações que facilitem a descoberta de conhecimento por pesquisadores e cientistas.

2 Pesquisas Relacionadas

Apesar da inovação trazida pela ferramenta proposta, ela se insere no contexto consolidado da bibliometria. O projeto dialoga com o ecossistema de aplicações já existentes na área, visando não apenas complementar, mas também expandir os recursos de análise atualmente à disposição dos pesquisadores.

2.1 Bibliometria

A bibliometria, também conhecida como infometria, ou cientometria, é uma área da ciência da informação que utiliza métodos quantitativos para analisar a produção e a disseminação do conhecimento científico (Lawani, 1981; Otlet, 1934). Segundo Price (1969), a cientometria é o estudo quantitativo da atividade científica.

Por meio de técnicas estatísticas e matemáticas, a bibliometria estuda publicações, como artigos, livros e patentes, permitindo medir o volume de publicações em determinadas áreas ou por autores específicos, além de avaliar o impacto das publicações com base no número de citações recebidas (Fidelis *et al.*, 2009). A Figura 2 mostra um exemplo de rede de coautoria, numa ferramenta *VOSviewer* (van Eck & Waltman, 2010) muito conhecida no campo de cientometria.

A análise estatística de informações bibliográficas e a formulação de modelos ou leis têm sido realizadas desde o século XIX, de acordo com Boustany (1997). De forma sistemática teve início no século XX, com os estudos de Lotka (Voos, 1974). Desde então, as informações bibliográficas, organizadas em bancos de dados públicos, acessíveis gratuitamente ou mantidos por serviços comerciais, tornaram-se objeto de numerosos estudos, que resultaram em novas denominações, dependendo do foco da pesquisa: cientometria, infometria, tecnometria, museometria, arquiometria, iconometria, biblioteconometria, webmetria, entre outras (Rostaing, 1996).

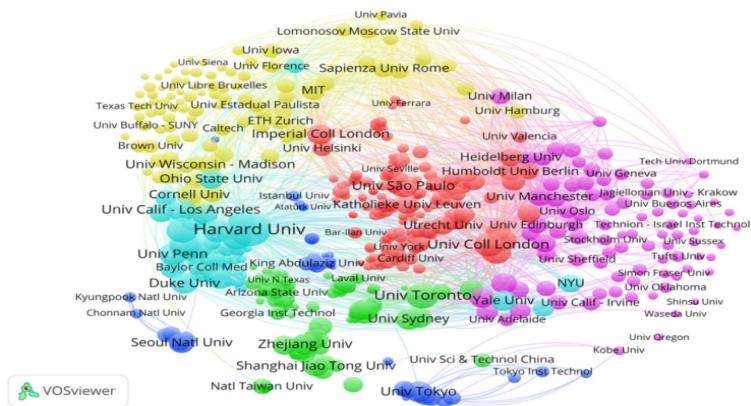


Figura 2: Um exemplo de visualização da rede de coautoria universitária, na ferramenta bibliométrica *VOSviewer*, construída por meio de contagem completa disponível em Perianes-Rodriguez *et al.* (2016). Uma visualização interativa está disponível em van Eck & Waltman (2010).

2.2 Ferramentas semelhantes

Existem outras ferramentas que possuem propostas semelhantes à Summarticles, como o *VOSviewer* (Van Eck & Waltman, 2013), que se trata de uma ferramenta gratuita para visualização e análise de redes científicas, muito semelhante ao *Gephi* (Khokhar, 2015). O *VOSviewer* permite a criação de gráficos de rede, visualização de redes de coautoria, mapas de calor, entre outras visualizações de dados. Contudo, o *VOSviewer* necessita de dados estruturados. Seguindo a mesma abordagem de *VOSviewer*, temos *SciMAT* (Cobo *et al.*, 2012) e *CiteSpace* (Synnestvedt *et al.*, 2005), que, para sua utilização, exigem que os dados dos artigos científicos estejam bem estruturados para realizar suas análises, o que torna o processo moroso na preparação dos dados.

Neste sentido, há também *Bibliometrix* (Aria & Cuccurullo, 2017), que se trata de um pacote de software escrita na linguagem R, com o objetivo de realizar análises bibliométricas, como a análise de coautoria, análise de citação por artigo, análise de citação por autor, entre outras análises bibliométricas. É necessário saber programar na linguagem R para utilizá-la, e os dados dos artigos científicos devem estar bem estruturados para a análise.

Sciswing (Kashyap & Kan, 2020), um pacote Python, pode ser usado para extração de dados, execução de modelos de aprendizado de máquina e recuperação de informações científicas de artigos. Porém, é preciso pré-processar os dados e ter conhecimento na linguagem Python.

Temos também a ferramenta Leximancer Pty Ltd (2005-2026), com proposta semelhante a ferramenta Scholarcy (2026), que realiza análise de texto com técnicas de processamento de linguagem natural. Com esta ferramenta, é possível realizar a extração de palavras chave em textos, análise semântica, analisar relações de dependência e tendências nos padrões dos dados, além de gerar visualizações, como mapas de calor, redes de dependência e frequência. Ela é uma ferramenta robusta; porém, é uma ferramenta de análise de dados não gratuita.

Para a utilização da ferramenta proposta neste projeto, não há necessidade de saber nenhuma linguagem de programação, não requer uma preparação prévia dos dados textuais de artigos científicos, e trata-se de uma ferramenta *open source* e gratuita. Contudo, é necessária a instalação e configuração de algumas plataformas como o *Docker*, *Ollama* e *Python*.

3 Fundamentação Teórica

3.1 O Processo de Descoberta de Conhecimento

O processo de descoberta de conhecimento fundamenta-se em identificar, receber e poder processar informações relevantes e agregá-las ao conhecimento prévio de seu usuário, mudando o estado de seu conhecimento atual, a fim de que determinada situação ou problema possa ser resolvido, como descrito por Wives (2002), e está fortemente relacionado com à forma pela qual a informação é processada (Morais & Ambrósio, 2007). Neste projeto de pesquisa, essa relação é direta e necessária para atender aos seus objetivos.

O volume de informações e dados disponíveis é muito grande e vem crescendo ao longo dos anos com o avanço da tecnologia e do *bigdata* (Gupta & Rani, 2019). Nesse sentido, o volume das publicações acadêmicas e artigos científicos aumentou consideravelmente ao longo dos anos (Fire & Guestrin, 2019).

Diante deste cenário, faz-se necessário facilitar o processo de descoberta de conhecimento por meio de automatizações, *software* que possam processar um grande volume de dados dos mais variados tipos, juntando e processando dados estruturados, semi-estruturados e não estruturados.

3.2 Tipos de Dados

Os dados estão distribuídos em três tipos distintos: dados estruturados, dados semi-estruturados e dados não estruturados.

3.2.1 Dados Estruturados

Os dados estruturados são informações que seguem uma estrutura definida ou padrão, como tabelas, formulários ou outras estruturas convencionais de dados. Eles são organizados de maneira clara e precisa, facilitando o processamento, a análise e a recuperação de informações (Doan *et al.*, 2009).

Exemplos de dados estruturados incluem informações em tabelas de bancos de dados,

registros em arquivos tabulares, como arquivos CSV ou planilhas, como na [Figura 3](#), entre outros tipos. Estes dados são geralmente processados e armazenados de maneira mais eficiente do que dados não estruturados, pois seguem uma estrutura bem definida.

Na academia científica, a análise de dados estruturados é muito importante, pois a maioria dos dados de experimentos científicos é estruturada e bem definida. Isso facilita a reprodutilidade, o armazenamento e a compreensão pelos cientistas ([Wong et al., 2010](#)).

sepal.length	sepal.width	petal.length	petal.width	variety
5.1	3.5	1.4	.2	Setosa
4.9	3	1.4	.2	Setosa
4.7	3.2	1.3	.2	Setosa
4.6	3.1	1.5	.2	Setosa
5	3.6	1.4	.2	Setosa
5.4	3.9	1.7	.4	Setosa
4.6	3.4	1.4	.3	Setosa
5	3.4	1.5	.2	Setosa
4.4	2.9	1.4	.2	Setosa
4.9	3.1	1.5	.1	Setosa
5.4	3.7	1.5	.2	Setosa

Figura 3: Um pequeno conjunto de dados estruturados clássico de [Fisher \(1936b\)](#). Um dos primeiros conjuntos de dados conhecidos usados para avaliar métodos de aprendizado de máquina na tarefa de classificação, disponível em [Fisher \(1936a\)](#).

3.2.2 Dados Semi-Estruturados

Os dados semi-estruturados são aqueles que possuem uma estrutura parcial ou implícita, ou seja, eles não seguem uma estrutura completamente definida, mas, simultaneamente, não são completamente não estruturados ([Hänig et al., 2010](#)). Eles podem ser representados como uma mistura de dados estruturados e não estruturados.

Exemplos de dados semi-estruturados incluem arquivos XML, HTML, JSON e arquivos de texto que contêm informações organizadas de forma não completamente padronizada, na [Figura 4](#) temos um exemplo de arquivo XML semi-estruturado. Estes tipos de dados são mais fáceis de processar e analisar do que dados não estruturados, mas ainda requerem alguma forma de processamento para extrair informações úteis.

```

<?xml version="1.0" encoding="UTF-8"?>
<TEI xml:space="preserve" xmlns="http://www.tei-c.org/ns/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.tei-c.org/ns/1.0 https://raw.githubusercontent.com/kermitt2/grobid/master/grobid-home/schemas/xsd/Grobid.xsd"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  <teiHeader xml:lang="en">
    <fileDesc>
      <titleStmt>
        <title level="a" type="main" xml:id="_td4Fmmd">Offline A/B testing for Recommender Systems</title>
      </titleStmt>
      <publicationStmt>
        <publisher/>
        <availability status="unknown"><licence/></availability>
      </publicationStmt>
      <sourceDesc>
        <biblStruct>
          <analytic>
            <author>
              <persName><forename type="first">Alexandre</forename><surname>Gilotte</surname></persName>
              <affiliation key="aff0">
                <note type="raw_affiliation">Criteo Research</note>
                <orgName type="department">Criteo Research</orgName>
              </affiliation>
            </author>
            <author>
              <persName><forename type="first">Clément</forename><surname>Calauzènes</surname></persName>
              <affiliation key="aff0">
                <note type="raw_affiliation">Criteo Research</note>
                <orgName type="department">Criteo Research</orgName>
              </affiliation>
            </author>
            <author>

```

Figura 4: Um exemplo de artigo pré-processado pela ferramenta Summaarticles e transformado em um arquivo do tipo .xml semi-estruturado.

3.2.3 Dados Não Estruturados

Dados não estruturados são informações que não seguem uma estrutura bem definida ou padrão, como tabelas, formulários ou outras estruturas convencionais de dados (Feldman & Sanger, 2007). Estes tipos de dados incluem informações como texto livre, imagens, áudios, vídeos, mensagens de correio eletrônico, entre outros. Os dados não estruturados são geralmente mais complexos de processar e analisar do que os dados estruturados, pois não seguem uma forma definida.

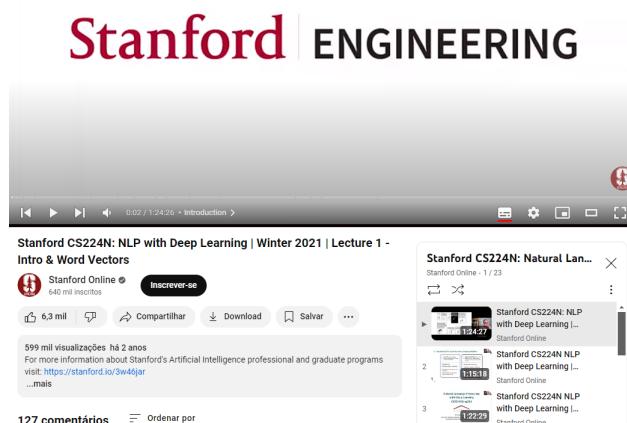


Figura 5: Um exemplo de dado não estruturado são vídeos e imagens. A imagem mostra uma playlist do YouTube que possui dados semi-estruturados e não estruturados, disponível em Stanford Online (2021).

3.3 Artigo Científico

Um artigo científico é um documento acadêmico que descreve os resultados de uma pesquisa científica realizada por especialistas em uma área do conhecimento. Esses artigos são publicados em anais de eventos, revistas ou periódicos, que são repositórios de divulgação destinados a compartilhar informações científicas. Artigos científicos normalmente trazem hipóteses que precisam ser avaliadas, testadas e desafiadas ao longo do documento (Keshav, 2007).

”A comunicação oral ou escrita tem começo, meio e fim. O autor introduz o tema, desenvolve-o e conclui. A redação do artigo científico acompanha essa mesma sequência. Dentre as partes essenciais que a compõem, encontram-se as informações básicas sobre o assunto investigado, a justificativa para o estudo e um claro objetivo, aquele que o investigador se propôs a alcançar com a realização da pesquisa ou no relato dos seus resultados. São ainda arrolados os fatos e argumentos, em ordem lógica, para convencer o leitor de que a conclusão está devidamente fundamentada. A conclusão representa a resposta do autor ao objetivo da investigação, ligando-se o desfecho com a questão que motivou a pesquisa. Não importa o tipo de artigo, seja original ou revisão, pesquisa qualitativa ou quantitativa, de cunho experimental ou não, o texto segue o mesmo encadeamento de ideias.” (Ribeiro, 2016, p.70-71)

Os artigos científicos seguem um formato estruturado e são escritos de maneira clara e objetiva. A [Figura 6](#) descreve uma estrutura de artigo científico.

Seção	Conteúdo	Pergunta-chave
Introdução	Apresentação de informações sobre o tema, a justificativa para a investigação e o objetivo.	De que trata o estudo? Por que a investigação foi feita? O que se sabia sobre o assunto?
Método	Descrição do cenário da pesquisa, da amostra, dos procedimentos e dos aspectos éticos.	Como o estudo foi realizado?
Resultados	Apresentação dos achados acompanhados, se aplicável, da respectiva análise estatística.	O que foi encontrado? Quais são os fatos revelados pela investigação?
Discussão	Interpretação dos resultados, comparações e conclusão.	O que significam os achados apresentados? O que este estudo acrescenta ao que já se sabia sobre o assunto?

Figura 6: Estrutura IMRD do artigo científico de acordo com Ribeiro (2016).

Os artigos geralmente consistem em seções como introdução, revisão da literatura, metodologia, resultados, discussão e conclusões. Normalmente, os artigos científicos seguem a estrutura IMRD: introdução, método, resultados e discussão (Ribeiro, 2016).

Os artigos científicos são submetidos a um processo de revisão por pares, no qual outros especialistas na área analisam o trabalho antes e depois de sua publicação. Isso ajuda a garantir a qualidade e a validade dos resultados apresentados (Ribeiro, 2016, p.76). Além disso, os

artigos científicos devem seguir normas e diretrizes (ICMJE *et al.*, 2006) específicas de cada revista científica, como formatação, estilo de escrita e referências bibliográficas (Pasquarelli, 2004; Wainer *et al.*, 2007).

A publicação de artigos científicos desempenha um papel fundamental na comunicação e na progresso da ciência, permitindo que pesquisadores compartilhem suas descobertas, ampliem o conhecimento existente e inspirem novas investigações (Meadows & de Lemos Lemos, 1999; Mueller, 1995). Ademais, a leitura de artigos científicos possibilita que outros pesquisadores se mantenham atualizados sobre os avanços em suas áreas de interesse e fundamentem suas próprias pesquisas em evidências científicas (Ribeiro, 2016, p.68-70).

Existem algumas diretrizes mencionadas na literatura científica que são utilizadas por pesquisadores, como a CONSORT, usada para relatos de ensaios clínicos (Schulz *et al.*, 2014); a PRISMA, usada para relatos de revisão sistemática e metanálise de estudos randomizados (Selçuk, 2019); a MOOSE, para relatos de metanálise de estudos observacionais em epidemiologia (Brooke *et al.*, 2021); o STARD, utilizado para relatos de estudos de diagnóstico (Cohen *et al.*, 2016); o STROBE, para relatos de estudos de corte transversal e caso-controle (Von Elm *et al.*, 2007); e o TREND, que provê um guia de boas práticas para relatar intervenções não aleatórias (Haynes *et al.*, 2021). Essas diretrizes que guiam alguns tipos de produções científicas são produto de boas práticas de publicações, guias que podem orientar os pesquisadores e que possuem consenso entre boa parte da comunidade científica. Contudo, essas diretrizes não são regras, e são apenas guias de recomendações para os interessados em comunicação científica.

Neste projeto, tratamos os artigos científicos como documentos textuais. Isto é, dados não estruturados do tipo textual. Para fins deste trabalho, elementos não textuais como figuras e tabelas não são considerados. Os artigos passam por transformações e processamento de dados, para tornarem-se dados estruturados.

Um artigo científico é composto por uma sequência de palavras, símbolos e pontuações, também chamadas de *tokens* (Wetzel, 2018), pode conter também diagramas, imagens, representações matemáticas, entre outros tipos de informações. A maior parte da informação contida em um artigo científico está na forma textual, representando um corpo textual; entretanto, é importante entender o que é um artigo científico, sua estrutura sintática e semântica.

3.3.1 Estrutura de um Artigo Científico

Um artigo científico geralmente segue uma estrutura organizada e padronizada para facilitar a compreensão e o acesso aos resultados da pesquisa. Embora possam existir variações dependendo da área de estudo e das diretrizes para publicação em periódicos, a estrutura básica de um artigo científico inclui as seguintes seções:

Título identifica um artigo científico, é usado para buscá-lo e ajuda a indexar um artigo científico. O título deve ser conciso, descritivo e refletir o conteúdo do artigo.

"Selecionar o título de um artigo é uma tarefa aparentemente fácil, mas primeiro pense bem no tema do seu trabalho. Analise títulos semelhantes e tente identificar facilmente o seu entre eles.

O título deve ter as seguintes características:

1. Seja atraente, de modo que descreva o conteúdo do artigo de forma específica, clara, precisa, breve e concisa.
2. Permita que o leitor identifique o tópico facilmente.
3. Permitir indexação precisa do material.

O autor tem 3 oportunidades para escolher, modificar, alterar o título do artigo:

1. antes de começar o trabalho.
2. no decorrer da escrita.
3. no final do artigo.

Se após essas 3 oportunidades você não conseguir um bom título, seu artigo terá poucos leitores e quem o ler irá criticá-lo mentalmente." **Jara Casco (1999)**

Resumo é um breve apanhado do estudo, destacando o objetivo, os métodos, os principais resultados e as conclusões (**Koopman, 1997**). É uma seção importante, pois muitas pessoas leem apenas o resumo para decidir se irão ler o artigo completo (**Lancaster, 2004**). Segundo **Ribeiro (2016)**, o resumo do artigo científico serve para comunicar uma visão concisa do documento, destacar pontos relevantes ou inovadores da pesquisa, ajudar o leitor a decidir se prossegue ou não com a leitura do artigo, auxiliar o leitor a recordar as características principais da pesquisa e facilitar a organização do plano para a redação da primeira minuta do texto.

Introdução estabelece o contexto do estudo, apresentando o problema de pesquisa, destacando a importância do tema e fornecendo uma revisão da literatura relevante (**Pereira, 2012**). De acordo com o manual de elaboração do artigo de **Joseilme Fernandes Gouveia (2017)** "A

introdução deve incluir contextualização, relevância e justificativa da pesquisa, fundamentada em referências relacionadas ao objetivo do artigo, descrição do problema, objetivo geral e contribuições”.

Essa seção do artigo prepara o leitor para compreender o estudo e a justificativa de sua execução. Deve ser estruturada de maneira a despertar o interesse do leitor e incentivá-lo a continuar a leitura. A introdução de um artigo científico é importante porque pode revelar a conclusão do artigo (Gastel & Day, 2022).

”When entering a gate of a magnificent city we can make a prediction about the splendor, pomposity, history, and civilization we will encounter in the city. Occasionally, gates do not give even a glimpse of the city, and it can mislead the visitors about inner sections of the city. Introduction sections of the articles are like gates of a city. It is a presentation aiming at introducing itself to the readers, and attracting their attention.” Armağan (2013)

Metodologia ou **método** descreve em detalhes como a pesquisa foi conduzida. Isso inclui informações sobre a amostra ou participantes, os procedimentos adotados, os instrumentos utilizados e as análises estatísticas realizadas. Essa seção deve ser escrita de forma suficientemente detalhada para que outros pesquisadores possam replicar o estudo.

”[...] método comprehende o material e os procedimentos adotados na pesquisa de modo a poder responder à questão central da investigação. Inclui, dentre outros, o tipo de delineamento, a forma de seleção dos indivíduos para compor a amostra do estudo, a maneira de coletar dados e de analisá-los. Os procedimentos se tornaram, com o passar do tempo, complexos e cientificamente mais válidos, o que fez com que essa seção do artigo possa estar repleta de informações especializadas.” Ribeiro (2016)

Resultados apresentam os principais achados da pesquisa (Gastel & Day, 2022, p.77-79). Geralmente, os dados coletados são apresentados de forma objetiva, utilizando tabelas, gráficos ou figuras. Os resultados devem ser descritos de maneira clara e objetiva, sem interpretações naquela seção.

Discussão apresenta os resultados, os quais são interpretados e contextualizados em relação à pergunta de pesquisa e à literatura existente (Gastel & Day, 2022, p.81-83). Os autores analisam os achados, discutem suas implicações e limitações e sugerem direções para pesquisas futuras.

Conclusões resumem os principais pontos do estudo e respondem à pergunta da pesquisa. Elas devem ser baseadas nos resultados apresentados e fornecer uma visão geral clara e concisa

do estudo (Gastel & Day, 2022, p.84).

Referências incluem todas as fontes citadas no artigo (Gastel & Day, 2022, p.89-97). Deve seguir um formato de citação específico, como as normas da ABNT (Associação Brasileira de Normas Técnicas) (Pasquarelli, 2004), SBC (Sociedade Brasileira de Computação), APA (*American Psychological Association*), MLA (*Modern Language Association*) e outras.

Além dessas seções principais, um artigo científico pode conter apêndices, agradecimentos e informações sobre financiamento, dependendo das normas da revista científica em que será publicado (Ribeiro, 2016). Cada seção tem um papel importante na estrutura do artigo científico e ajuda a apresentar os resultados da pesquisa de forma clara e acessível aos leitores.

3.4 Processamento de Linguagem Natural

Uma linguagem natural refere-se a um dos modos como os seres humanos se comunicam, seja por meio de palavras escritas ou faladas (Locke & Bogin, 2006; Wildgen, 2004). Quando adicionamos o termo processamento, a linguagem natural passa a ser objeto de estudo em meios tecnológicos digitais:

”Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications.” Chowdhary (2020)

Os computadores não conseguem compreender ou interpretar a linguagem falada ou escrita. Por esse motivo, temos que representar essas informações humanas de uma forma que seja possível o seu processamento pelos computadores (Cerf, 1969; Harris, 1954b; Hayles, 2010; Pilehvar & Camacho-Collados, 2020). Esse é o grande objetivo do processamento de linguagem natural, que busca ajudar o computador a realizar o reconhecimento e a compreensão da linguagem humana para executar inúmeras tarefas, como: geração automática de texto, classificação textual (Paiva *et al.*, 2021), diálogo e interações (Gatzioufa & Saprikis, 2022), extração de informações (Kumar *et al.*, 2007; Park *et al.*, 2019), sumarização de textos, tradução automática (Hassan *et al.*, 2018; Singh *et al.*, 2017), pesquisa em textos (Kobayashi & Takeda, 2000), análise de sentimento (Yadav & Vishwakarma, 2020), entre tantas outras tarefas.

”The idea of giving computers the ability to process human language is as old as the idea of computers themselves. [...] vibrant interdisciplinary field with many names corresponding to its many facets, names like speech and language processing, human language technology, natural language processing, computational linguistics, and speech recognition and synthesis. The goal of this new field is to get computers to perform useful tasks involving human language, tasks like enabling human-machine communication, improving human-human communication, or simply doing useful processing of text or speech.” Jurafsky & Martin (2024)

O campo de processamento de linguagem natural (PLN) teve seu início na década de 1950, com os trabalhos de Booth *et al.* (1955) na tradução automática da língua russa para a língua inglesa. A maior parte da pesquisa feita em PLN durante este período inicial do campo de pesquisa concentrou-se em técnicas de análise sintática, em parte porque o processamento sintático era mais necessário e menos desafiador quando comparado a incorporação de técnicas que analisavam a semântica textual, como descreve Jones (1994). Na década de 1960, Joseph Weizenbaum desenvolveu o programa ELIZA no Massachusetts Institute of Technology (MIT) em 1966, e foi um dos primeiros programas de IA a incorporar PLN (O'Regan, 2018). Nas décadas seguintes, em paralelo com a evolução tecnológica, o PLN continuou progredindo, incorporando novas técnicas de processamento de dados e inteligência artificial (Jurafsky & Martin, 2024, p.9-14).

”This is an exciting time for the field of speech and language processing. The startling increase in computing resources available to the average computer user, the rise of the Web as a massive source of information, and the increasing availability of wireless mobile access have all placed speech- and language-processing applications in the technology spotlight.” Jurafsky & Martin (2024)

A grande parte da informação que transita entre os meios digitais tecnológicos é textual (Cukier, 2010). Portanto, é crucial ter estratégias e métodos para o processamento de um grande volume de informações textuais (Manning, 1999). À medida que a quantidade de dados continua a crescer (Sagiroglu & Sinanc, 2013) e a necessidade de interação com máquinas aumenta, o PLN desempenha um papel cada vez mais importante na sociedade e em diversos setores, impulsionando a automação em larga escala (Chowdhary, 2020).

Neste projeto, artigos científicos são documentos textuais. Um documento contendo uma estrutura bem definida, com inúmeras palavras, imagens, gráficos e tabelas. Será utilizada somente a informação textual dos artigos científicos para análises de PLN.

Um documento é composto por palavras, e as palavras são uma junção de letras e sílabas, isto é, palavras são uma sequência de letras. Uma palavra pode ser entendida como um *token* (Wetzel, 2018) ou uma junção de *tokens* (Zouhar *et al.*, 2023). Um documento textual contém informação textual léxica e informação semântica (sentido, contexto e ordem de aparição das palavras).

3.4.1 Pré-processamento de Textos

Dados em formato textual são dados não estruturados, dados que não possuem uma estrutura definida. Por conta da estrutura não definida, trabalhar com dados textuais impõe outra dificuldade, pois existe a necessidade de extrair, processar, tratar e representar numericamente o texto antes de processá-lo (Manning, 1999).

O pré-processamento de texto refere-se a uma série de processos ou técnicas aplicadas a um conjunto de dados textuais antes de serem utilizados para análise ou processamento adicional. Essas etapas são realizadas com o objetivo de limpar, normalizar e preparar os dados textuais de forma adequada, a fim de obter melhores resultados nas tarefas de processamento de linguagem natural.

Normalmente, dependendo das necessidades específicas da tarefa ou do tipo de análise a ser realizada, várias etapas ocorrem no pré-processamento de texto. Por exemplo, tarefas como tradução automática ou detecção de similaridade entre documentos podem necessitar de um tratamento do dado textual de formas distintas.

Alguns conceitos são fundamentais na definição e no entendimento da estrutura dos dados textuais. A seguir, veremos conceitos e definições para o tratamento de dados textuais. Na [Tabela 1](#) apresentamos exemplos desses conceitos.

Segundo Webster & Kit (1992), podemos entender *tokens* como palavras. Contudo, isso pode não se aplicar a todas as situações:

"The entity word is one kind of token for NLP, the most basic one. Our concern, however, is with using the computer to recognize those tokens without distinct delimiters, such as Chinese words, English idioms and fixed expressions. [...] It is believed that, by taking idioms and fixed expressions as a kind of basic unit at the same level as words, tokenization should take on a more generalized and realistic significance making NLP and MT systems more robust and practical." Webster & Kit (1992)

Tokens, neste projeto, podem ser entendidos como as unidades básicas em que um dado textual pode ser dividido para análise. Essas unidades básicas podem ser palavras, pontuações, radicais, subpalavras ou caracteres (símbolos, pontuações ou alguns tipos de caracteres especiais), dependendo da abordagem utilizada.

Types são os *tokens* únicos em um conjunto de dados textuais. Eles compõem o vocabulário quando ignoramos pontuações e caracteres especiais, temos as instâncias de palavra. Corpus é um conjunto estruturado de dados textuais ou dados linguísticos. Corpora é o plural de corpus, e refere-se a múltiplas coleções de dados textuais ou dados linguísticos.

A normalização trata-se da padronização dos *tokens* para uma forma comum e simplificada (Bird *et al.*, 2009, c.3.6). A normalização de textos costuma ter três tarefas associadas: tokenização, padronização das palavras e segmentação das sentenças (Jurafsky & Martin, 2024, p.17). Na [Tabela 3](#) podemos ver um exemplo de normalização de dados textuais.

Tokenização é o processo de dividir os dados textuais em *tokens*. Um exemplo pode ser visto na [Tabela 2](#). Não podemos analisar o dado textual em sua estrutura bruta e não estruturada. Um caminho para realizar a análise do dado textual é quebrá-lo em partes menores, a fim de torná-lo mais estruturado. O processo de tokenização transforma um texto contínuo, que está disposto em uma sequência de caracteres, em uma lista de *tokens* que podem ser facilmente manipulados e analisados por técnicas estatísticas ou algoritmos de PLN (Webster & Kit, 1992).

Na [Tabela 2](#) podemos ver a tokenização da seguinte frase do livro Dom Casmurro de de Assis (1899):

”A imaginação foi a companheira de toda a minha existência, viva, rápida, inquieta, alguma vez tímida e amiga de empacar, as mais delas, capaz de engolir campanhas e campanhas, correndo.”

Tabela 1: Exemplo de aplicação dos conceitos de processamento de linguagem natural. Utilizando uma passagem do livro De Assis (1998).

Corpora	Todos os dados linguísticos da literatura brasileira.
Corpus	Todos os dados linguísticos dos livros de Machado de Assis
Frase	Ao verme que primeiro roeu as frias carnes do meu cadáver dedico como saudosa lembrança estas memórias póstumas.
tokens	[“Ao”, “verme”, “que”, “primeiro”, “roeu”, “as”, “frias”, “carnes”, “do”, “meu”, “cadáver”, “dedico”, “como”, “saudosa”, “lembrança”, “estas”, “memórias”, “póstumas”, “.”]
Types	[“Ao”, “verme”, “que”, “primeiro”, “roeu”, “as”, “frias”, “carnes”, “do”, “meu”, “cadáver”, “dedico”, “como”, “saudosa”, “lembrança”, “estas”, “memórias”, “póstumas”, “.”]
Instâncias	[“Ao”, “verme”, “que”, “primeiro”, “roeu”, “as”, “frias”, “carnes”, “do”, “meu”, “cadáver”, “dedico”, “como”, “saudosa”, “lembrança”, “estas”, “memórias”, “póstumas”]

Tabela 2: Tokenização de frase do livro Dom Casmurro de de Assis (1899).

A imaginação foi a companheira de toda a minha existência, viva, rápida, inquieta, alguma vez tímida e amiga de empacar, as mais delas, capaz de engolir campanhas e campanhas, correndo.	[“A”, “imaginação”, “foi”, “a”, “companheira”, “de”, “toda”, “a”, “minha”, “existência”, “”, “viva”, “”, “rápida”, “”, “inquieta”, “”, “alguma”, “vez”, “tímida”, “e”, “amiga”, “de”, “empacar”, “”, “as”, “mais”, “delas”, “”, “capaz”, “de”, “engolir”, “campanhas”, “e”, “campanhas”, “”, “correndo”, “.”]
---	---

Tabela 3: Exemplo de normalização de frase do livro Dom Casmurro de de Assis (1899)

Dados textuais	A imaginação foi a companheira de toda a minha existência, viva, rápida, inquieta, alguma vez tímida e amiga de empacar, as mais delas, capaz de engolir campanhas e campanhas, correndo.
Normalizado	a imaginacao foi a companheira de toda a minha existencia viva rapida inquieta alguma vez timida e amiga de empacar as mais delas capaz de engolir campanhas e campanhas correndo
Types	[“a”, “imaginação”, “foi”, “companheira”, “de”, “toda”, “minha”, “existencia”, “viva”, “rapida”, “inquieta”, “alguma”, “vez”, “timida”, “e”, “amiga”, “empacar”, “as”, “mais”, “delas”, “capaz”, “engolir”, “campanhas”, “correndo”]

Tabela 4: Exemplo de aplicação dos conceitos de stemização e lematização.

Palavras	Stem	Lema
corro, corres, corre, corremos, correis, correm, correra, correras, correra, corrêramos, corrêreis, correram, correria, correrão, corri, correste	corr	correr

Para aplicar a tokenização existem alguns algoritmos, como discorre (Jurafsky & Martin, 2024, p.18-23). Esses algoritmos variam desde uma simples expressão regular até algoritmos mais complexos, como o top-down tokenization, Penn Treebank tokenization, Chinese Treebank tokenization e Byte-Pair Encoding tokenization. A aplicação da tokenização sofre forte influência de diferentes línguas, pois, em alguns idiomas, o conceito de *token* pode variar e gerar ambiguidades (Hardeniya *et al.*, 2016; Thanaki, 2017).

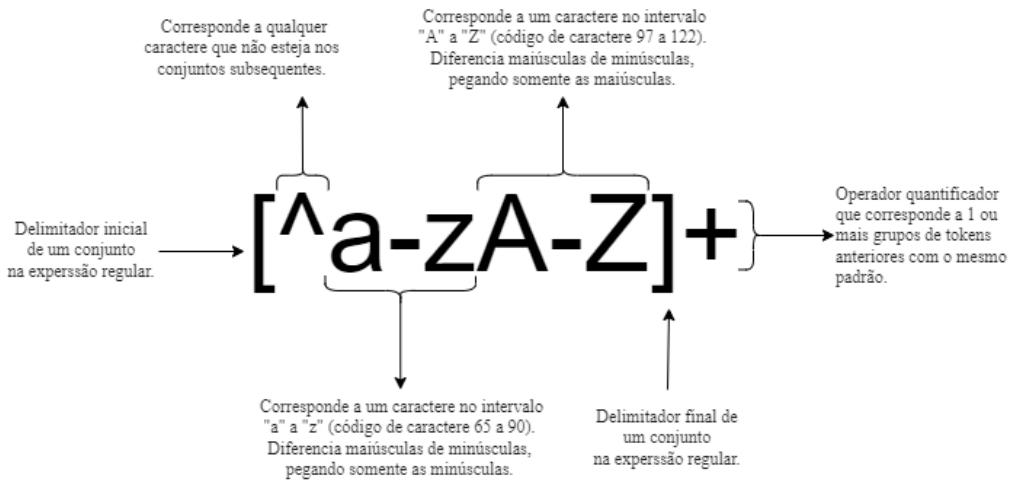


Figura 7: Exemplo de expressão regular usada na normalização de dados textuais em *Summaricles*.

A remoção de caracteres indesejados, como pontuação, símbolos e números, que podem não ser relevantes para a análise dos dados textuais, pode ser feita utilizando expressões regulares (regex) (Jurafsky & Martin, 2024, c.2) ou funções de substituição. Na Figura 7 temos um exemplo de regex usado para a remoção de caracteres não alfabéticos. A remoção dos caracteres ajuda na limpeza dos dados textuais, tornando-os menos complexos e mais estruturados para a análise de padrões ou a análise estatística. A remoção desses elementos desnecessários ajuda a focar na informação relevante, pois reduz a complexidade e a dimensionalidade dos dados.

Converter todas as palavras para letras minúsculas ou maiúsculas também é um tipo de normalização. Essa conversão é realizada com o intuito de melhorar a consistência textual e evitar a diferenciação de palavras. Em algumas línguas, como no português, há a presença de acentuação nas palavras. Nesta situação, um tratamento adicional é realizar a remoção da acentuação. Na Tabela 3 temos um exemplo desses processos.

Existem também outros tipos de normalização de textos, que são as padronizações de

palavras, como a lematização e stemização.

Lematização é a tarefa de reduzir uma palavra em todas as suas inflexões, retornando à sua forma base ou raiz, chamada de lema (Jurafsky & Martin, 2024, p.24). Segundo Lovins (1968), precursora na definição do conceito:

”[...] a stemming algorithm, a procedure to reduce all words with the same stem to a common form [...]”.

Por exemplo, a palavra ”correr” é o lema das seguintes palavras: correrá, correrão, correaram, entre outras variações. Percebemos que, independentemente de flexão, tempo verbal e mudanças relacionadas ao verbo ”correr”, todas as palavras tem em comum o mesmo lema. Logo, para simplificação e redução de complexidade, podemos usar o lema em vez de suas variações. Na [Tabela 4](#) temos um exemplo deste processo.

Stemização, segundo a definição e exemplo de Orengo & Huyck (2001):

”Stemming is the process of conflating the variant forms of a word into a common representation, the stem. For example, the words: “presentation”, “presented”, “presenting” could all be reduced to a common representation “present”. This is a widely used procedure in text processing for information retrieval (IR) based on the assumption that posing a query with the term presenting implies an interest in documents containing the words presentation and presented.” [Orengo & Huyck \(2001\)](#)

Para realizar a stemização, existem diversos algoritmos que variam de acordo com as características da linguagem e do idioma, por exemplo, para o inglês (Paice, 1990; Porter, 1980). Para a língua portuguesa, podemos citar Alvares *et al.* (2005); Orengo & Huyck (2001). Na [Tabela 4](#) temos um exemplo desse processo.

Stopwords, ou palavras de parada, são palavras muito comuns em um corpus textual. Porém, tais palavras não trazem nenhuma informação relevante sobre o dado textual. Por exemplo, artigos como ”o”, ”a”, ”os”, ”as”, ”um”, ”uns”; preposições como ”a”, ”em”, ”para”, ”por”; conjunções como ”mas”, ”ora”, ”ou”, ”então”; entre tantas outras palavras de parada.

De acordo com Aranha (2007), as palavras de parada (stopwords) são termos frequentes que aparecem nos mais variados dados textuais e não agregam informação relevante para a descoberta de conhecimento. Segundo Feldman & Sanger (2006), o intuito do pré-processamento é obter uma representação do dado textual em um formato mais estruturado, que preserve as principais características dos dados textuais originais.

Artigos, conjunções, preposições e pronomes, embora sejam fundamentais na construção de sentenças, não auxiliam na discriminação de texto por aparecerem em praticamente todos os documentos. A eliminação desses termos reduz o espaço necessário de armazenamento sem afetar a qualidade do sistema (Salton, 1983).

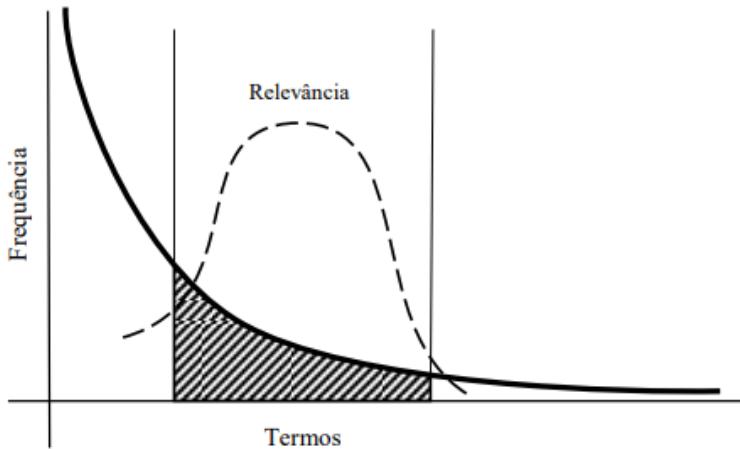


Figura 8: Diagrama de frequência das palavras. Quanto mais frequente ou extrema é uma palavra, menor é sua relevância em termos de informação relevante para compreensão do dado textual (Catae, 2012).

Na Figura 8 os termos estão dispostos de forma ordenada pela frequência das palavras. O trabalho de Luhn (1958) constatou que os termos mais relevantes não são aqueles que possuem alta frequência, nem os termos que têm baixa frequência. Dessa maneira, estabeleceu-se a existência de zonas de corte para definir onde estariam as palavras mais relevantes, e essas zonas seriam determinadas através de experimentos (Catae, 2012).

Contudo, o trabalho de Luhn foi realizado com base na Lei de Zipf (Piantadosi, 2014), que mostrou que uma das características das linguagens humanas e muitos outros fenômenos humanos e naturais, seguem uma distribuição semelhante, a qual foi denominada de *Principle of Least Effort* (Zipf, 2016).

N-gram é o tipo mais simples de modelo de linguagem pois, com esta abordagem, é possível relacionar as frequências de palavras em sequência, permitindo a aproximação da probabilidade de ocorrência de uma palavra dada a ocorrência de palavras anteriores (Jurafsky & Martin, 2024). De forma prática, n-grams são sequências de *tokens*, palavras, caracteres ou outros elementos extraídos de um texto (Cavnar et al., 1994).

Os n-grams são amplamente utilizados em processamento de linguagem natural

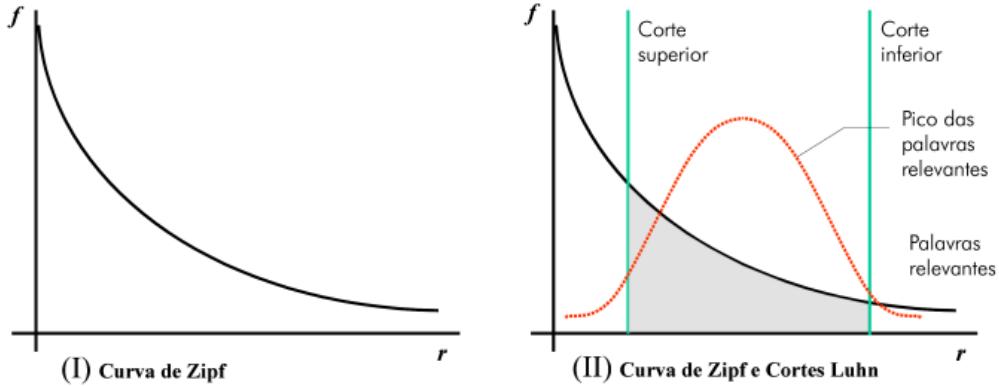


Figura 9: A curva de Zipf e os cortes de Luhn adaptados de Matsubara *et al.* (2003).

(PLN) para análise textual (Nessa *et al.*, 2008; Peng & Schuurmans, 2003; Reddy & Pujari, 2006), análise estatística (Lin & Hovy, 2003; Suen, 1979) e modelagem de linguagem (Tripathy *et al.*, 2016). Os n-grams podem ser classificados em unigramas (1-gram), bigramas (2-gram), trigramas (3-gram) ou, de maneira geral, n-grams, dependendo do número de itens na sequência de *tokens*.

Por exemplo, na frase "Summarticles sumariza um conjunto de artigos científicos", a aplicação do conceito estaria de acordo com a [Tabela 5](#).

Tabela 5: Exemplo de aplicação do conceito de n-grams.

Frase inicial	Summarticles sumariza um conjunto de artigos científicos.
Unigramas	["Summarticles", "sumariza", "um", "conjunto", "de", "artigos", "científicos", "."]
Bigramas	Unigramas + ["Summarticles sumariza", "sumariza um", "um conjunto", "conjunto de", "de artigos", "artigos científicos", "científicos ."]
Trigramas	Unigramas + Bigramas + ["Summarticles sumariza um", "sumariza um conjunto", "um conjunto de", "conjunto de artigos", "de artigos científicos", "artigos científicos."]

A ocorrência de *tokens* em sequência pode trazer informações sobre *tokens* ou palavras que possuem padrões de frequência relacionados. Por exemplo, a palavra "bom" tem muito mais chance de aparecer junto com a palavra "dia" do que com a palavra "noite". Os n-grams permitem extrair esses padrões de frequência conjunta entre *tokens* ou palavras (Kondrak, 2005).

Essas sequências têm diversas aplicações, como na modelagem de linguagem, onde ajudam a prever a próxima palavra com base nas anteriores (Siivola & Pellom, 2005), e na análise de sentimentos (Dey *et al.*, 2018), permitindo identificar padrões que indicam emoções

(Aisopos *et al.*, 2011). Embora os n-grams capturem informações contextuais valiosas, eles também apresentam desafios, como o aumento exponencial de combinações à medida que "n" cresce, o que pode resultar em problemas de dimensionalidade (Pauls & Klein, 2011).

3.5 Representação vetorial de dados textuais

Em algumas tarefas, como classificação de texto, similaridade de texto ou modelagem de tópicos, é necessário converter dados textuais em representações numéricas para que seja possível realizar análises e para que os algoritmos de inteligência artificial possam processá-los (Matsubara *et al.*, 2003). Pesquisadores buscaram formas de definir o significado de palavras em termos de vetores (Switzer, 1965).

Segundo Matsubara *et al.* (2003):

”A transformação dos documentos em uma representação mais adequada, como em uma tabela atributo-valor, é uma etapa de suma importância, visto que a representação desses documentos tem uma influência fundamental em quão bem um algoritmo de aprendizado poderá generalizar a partir dos exemplos”.

Isso é feito por meio de técnicas de vetorização, como o saco de palavras (*bag-of-words*) (Qader *et al.*, 2019) ou através de técnicas mais recentes, como as representações distribuídas de palavras, como as técnicas de *word embeddings* (Gutiérrez & Keith, 2019).

A publicação de Salton *et al.* (1975), do então artigo intitulado “A vector space model for automatic indexing”, introduziu a ideia de representação de documentos textuais através de vetores em um espaço de alta dimensão.

Embora Salton, na década de 1970, tenha introduzido a ideia de espaço de vetores na representação de dados textuais; em meados da década de 1950, os trabalhos de Firth (1957); Harris (1954a); Joos (1950); Osgood *et al.* (1957) buscavam representações matemáticas para o significado das palavras. Como descreve (Jurafsky & Martin, 2024, c.5-6) sobre estes trabalhos precursores:

”[...] to define the meaning of a word by its distribution in language use, meaning its neighboring words or grammatical environments. Their idea was that two words that occur in very similar distributions (whose neighboring words are similar) have similar meanings.” .

A [Figura 10](#) traz a noção vetorial para dados textuais e de que a relação de proximidade

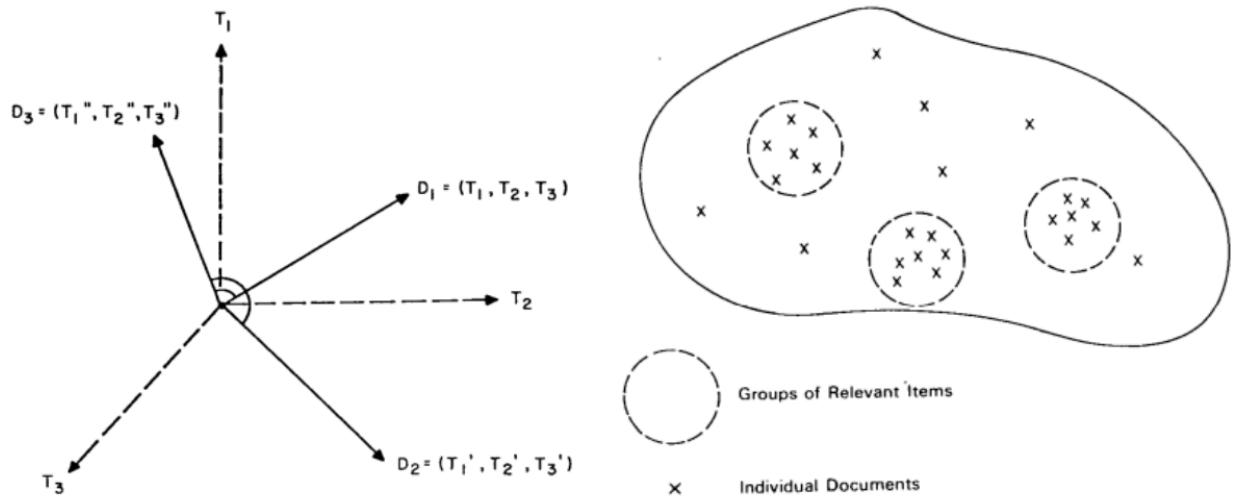


Figura 10: Representação do espaço vetorial de documentos D_i em função de seus termos T_i , adaptado de (Salton *et al.*, 1975).

entre vetores de dados textuais pode conter informações latentes e relações complexas não mapeadas.

3.5.1 Bag of Words (BOW)

Bag of Words é uma técnica de representação vetorial de textos. Trata-se de forma de converter texto em uma estrutura de fácil utilização para algoritmos. Com esta técnica, é possível representar qualquer documento textual como um vetor de frequências de seus *tokens*, essa representação recebe o nome de matriz de co-ocorrências ou matriz termo-documento (Jurafsky & Martin, 2024, c.6.3).

A técnica funciona da seguinte maneira: inicialmente, todo o corpus textual dos documentos é pré-processado e normalizado. Em seguida, o dado textual é dividido em palavras únicas, ou *types*, gerando um vocabulário de palavras relativo ao documento ou conjunto de documentos. Por fim, é contabilizada uma matriz de frequências que contém a frequência de cada palavra em cada documento (Santana, 2017).

Em um conjunto de documentos D , com k elementos e n palavras (termos ou *tokens*), representam-se os documentos D_1, D_2, \dots, D_k como vetores d_1, d_2, \dots, d_k no espaço vetorial $\mathbb{R}^n : d_j = (p_{1,i}, p_{2,i}, \dots, p_{n,i})$, para $1 \leq i \leq k$ em que $p_{1,i}, \dots, p_{n,i}$ são os pesos dos respectivos termos no documento D_i .

Nessa representação, o documento D_i é considerado um ponto no espaço de vetores de

dimensão n , como uma representação vetorial não ordenada de seus termos. A [Figura 10](#) exemplifica essa relação. A ordem relativa entre os termos no documento não é mantida, tornando-se necessário reter somente a informação de peso w_m do termo no documento D_i , que, por exemplo, pode ser a frequência do termo t_n no documento D_i . Por exemplo, as sentenças “ferramenta de sumarização de documentos é Summarticles” e “Summarticles é uma ferramenta de sumarização de documentos” não apresentam diferenças para esta representação. Por outro lado, as situações “a criança faz o cão feliz” e “o cão faz a criança feliz” possuem significados diferentes, embora sejam completamente idênticas no BOW.

Por exemplo, a [Tabela 6](#) mostra exemplos de documentos, e na [Tabela 7](#) temos esses mesmos documentos normalizados. Após o tratamento, aplica-se a extração do vocabulário, como pode ser visto na [Tabela 9](#) e a criação da matriz de frequências na [Tabela 8](#). Por fim, um caso geral da matriz de termo-documento e suas frequências pode ser visto na [Tabela 11](#), além desta estrutura matricial, é comum ter algoritmos que possuem como entrada a matriz transposta da [Tabela 11](#).

Tabela 6: Documentos que serão utilizando no *Bag of Words*

Documento	Texto
1	Summarticles é uma ferramenta para sumarização de textos científicos.
2	Summarticles facilita a vida de cientistas pois sumariza artigos científicos.

Tabela 7: Documentos normalizados para o *Bag of Words*

Documento	Texto
1º	summarticles ferramenta sumariza texto científico
2º	summarticles facilita vida cientista sumariza artigo científico

Tabela 8: Vocabulário do *Bag of Words*

Palavra única	Frequência
summarticles	2
ferramenta	1
sumariza	2
texto	1
científico	2
facilita	1
vida	1
cientista	1
artigo	1

Tabela 9: Matriz de frequência do *Bag of Words*

Palavra	Documento 1	Documento 2
summarticles	1	1
ferramenta	1	0
sumariza	1	1
texto	1	0
científico	1	1
facilita	0	1
vida	0	1
cientista	0	1
artigo	0	1

Tabela 10: Matriz de frequência do *Bag of Words*

Tabela 11: Matriz de frequência genérica do *Bag of Words*

Palavra única	Documento 1	...	Documento N
1 Palavra	2	...	5
...
N Palavra	0	...	1

3.5.2 TF-IDF

Embora as frequências sejam uma boa forma de representar o peso de um documento na representação *Bag of Word*, segundo Jurafsky & Martin (2024):

”[...] raw frequency is not the best measure of association between words. Raw frequency is very skewed and not very discriminative. If we want to know what kinds of contexts are shared by cherry and strawberry but not by digital and information, we’re not going to get good discrimination from words like the, it, or they, which occur frequently with all sorts of words and aren’t informative about any particular word”.

Como mostrado por (Luhn, 1958; Zipf, 2016), eventos mais frequentes nem sempre são mais relevantes, embora sejam muito observados na natureza.

De acordo com Buckley (1993):

”Evidence is presented that good weighting methods are more important than the feature selection process and it is suggested that the two need to go hand-in-hand in order to be effective.”.

Há diversas formas de definir a ponderação de um termo na representação vetorial de documentos, sendo que uma das mais conhecidas e utilizadas é o TF-IDF (Schütze *et al.*, 2008). Além dessa, existe a PPMI (Antoniak & Mimno, 2018).

Podemos definir o *TFIDF* como um arranjo matemático que incorpora o produto de duas componentes: *TF* (*term frequency*) e *IDF* (*Inverse Document Frequency*) (Jurafsky & Martin, 2024, c.6).

$$TFIDF = TF * IDF$$

$$TFIDF = tf_{t,d} * \log\left(\frac{N_d}{df_t}\right)$$

$$p_{t,d} = TFIDF = tf_{t,d} * \log\left(\frac{N}{df_t}\right)$$

Em que $tf_{t,d}$ representa o número de ocorrências do termo t no documento d , df_t corresponde ao número de documentos que possuem o termo t , e N é o total de documentos, com isto posto, podemos encontrar o valor de peso $p_{t,d}$.

Observe que o peso $p_{t,d}$ tende a aumentar quando o termo ocorre várias vezes em poucos documentos, diminui quando a frequência do termo é baixa, diminui quando o termo ocorre

em muitos documentos e torna-se nulo quando ocorre em todos os documentos (Catae, 2012). Portanto, podemos observar que $p_{t,d} = TF - IDF$ nos ajuda a medir termos que são mais relevantes para um documento, analisando a frequência com que aparecem em um documento, em comparação à sua frequência no conjunto total de documentos observados.

3.5.3 Vetores latentes (*embeddings*)

”It turns out that dense vectors work better in every NLP task than sparse vectors.” Jurafsky & Martin (2024)

Embeddings são vetores que representam palavras (Jurafsky & Martin, 2024). Contudo, quando abordamos o termo vetores latentes de palavras, não nos referimos às representações vetoriais esparsas e longas, como o BOW, estamos indicando tipos modernos de representações vetoriais curtas e densas, baseadas em análise de semântica latente (Deerwester *et al.*, 1990, 1989) ou arquiteturas de redes neurais artificiais (Bengio & Bengio, 1999; Bengio *et al.*, 2000; Mikolov, 2013). O termo ”*embedding*” foi usado inicialmente por Landauer *et al.* (1997) em um de seus trabalhos sobre análise de semântica latente (LSA).

Na representação vetorial BOW, dada uma coleção de documentos D , os vetores possuem tamanho k , denotando a quantidade t_k de *tokens* únicos. Entretanto, a maioria dos *tokens* tem um valor 0, resultando em uma alta dimensionalidade (maldição da dimensionalidade (Köppen, 2000)) e vetores esparsos (Bengio & Bengio, 1999). Na representação de vetores latentes por meio de embeddings, os vetores possuem tamanhos parametrizados (Deerwester *et al.*, 1990; Mikolov, 2013), de acordo com a necessidade de representação do problema, as dimensões não carregam uma interpretação clara (Jurafsky & Martin, 2024), porém seus valores observados no espaço de vetores carregam informações latentes que abstraem e capturam padrões dos dados (Mikolov, 2013).

Uma das deficiências das técnicas de matriz termo-documento frequência é a ausência da incorporação do contexto e da semântica das palavras. Por exemplo, quando há a ocorrência de palavras polissêmicas, com dois ou mais significados semânticos, o contexto deveria ser mapeado e levado em consideração (Catae, 2012).

Os modelos Word2Vec (Mikolov, 2013), FastText (Bojanowski *et al.*, 2017), GloVe (Pennington *et al.*, 2014) e BERT (Kenton & Toutanova, 2019), que são técnicas mais atuais

de conversão de palavras em vetores, revolucionaram a área de PLN quando foram lançados, devido à forma como cada uma dessas técnicas funciona. Na [Figura 11](#), podemos ver a diversidade de modelos que surgiram por meio dessas técnicas de representação vetorial de palavras.

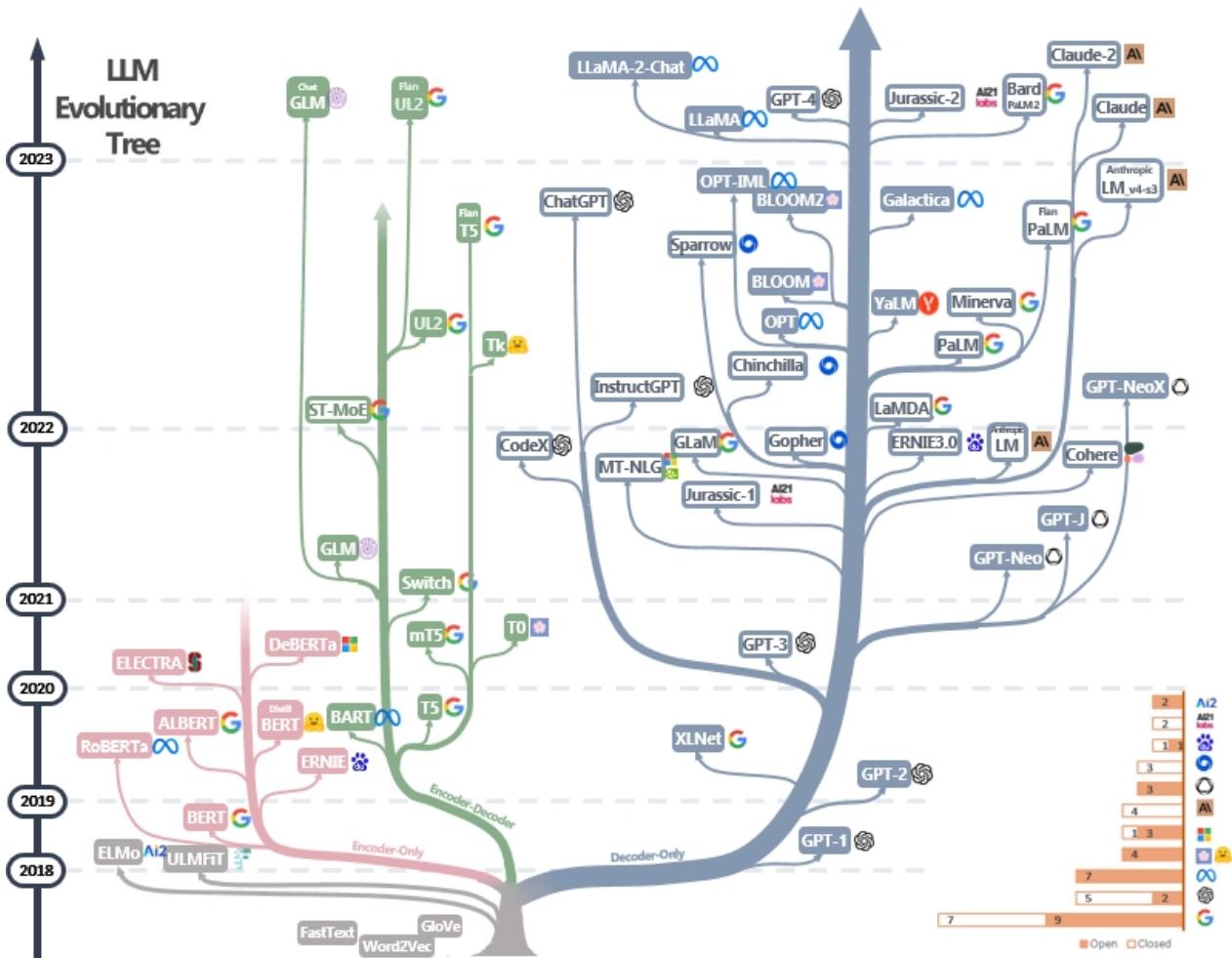


Figura 11: A árvore evolutiva dos *Large Language Models* (LLMs) modernos traça o desenvolvimento de modelos de linguagem, adaptado de Yang *et al.* (2024). A base destes modelos estão os modelos iniciais de embeddings, como por exemplo o Word2Vec (Mikolov, 2013).

Espaço Semântico (Latent Semantic Analysis, LSA) traz a noção de que as palavras em um dado texto possuem informações semânticas e contextuais, isto é, existem relações e informações implícitas que relacionam palavras no mesmo contexto (Deerwester *et al.*, 1990). A hipótese por trás do espaço semântico indica que existem conceitos ocultos entre as palavras, ou seja, todas as palavras estão conectadas indiretamente pelo contexto em que se encontram (Dumais *et al.*, 1988).

Segundo Catae (2012):

”A análise da semântica latente é uma técnica que projeta os vetores em um espaço de menor dimensão. Esse espaço de dimensão reduzida é denominado de espaço semântico, pois palavras similares ou relacionadas seriam projetadas em uma mesma dimensão do espaço”.

Vimos que as palavras podem ser representadas como vetores (Salton *et al.*, 1975) e que podemos fazer isso por meio do BOW (Qader *et al.*, 2019). Com esta representação em mãos, a estratégia do espaço semântico é evidenciar a semelhança entre os termos. Para isso, é preciso que, nesse novo espaço de vetores semântico, os termos mais próximos sejam agrupados, enquanto os termos distintos se afastem uns dos outros. Portanto, temos que sair de um espaço de vetores termo-documento para um novo espaço de vetores onde as relações de semântica se evidenciem, como apresentado formalmente por Papadimitriou *et al.* (1998).

A projeção dos vetores-documentos em um espaço de vetores semântico segmenta os termos e documentos automaticamente por similaridade, estabelecendo relacionamentos indiretos (Catae, 2012). O objetivo desta projeção é eliminar as dimensões irrelevantes, mantendo os vetores mais significativos e reduzindo a dimensão do espaço de vetores (esparsidade) (Deerwester *et al.*, 1990). Para a redução de dimensionalidade, é utilizada a técnica de decomposição de matrizes em valores singulares (SVD) (Dumais, 1991), que trata-se de um método para encontrar as dimensões mais importantes de um conjunto de dados, aquelas em que a variabilidade dos dados é maior (Henry & Hofrichter, 1992).

Em suma, com a matriz de frequência termo-documento, aplica-se a técnica de decomposição de matrizes em valores singulares, calcula-se a matriz de projeção resumida e, por fim, converte-se a matriz projetada nos vetores-documentos (Catae, 2012).

Word2Vec é uma técnica para produzir representações vetoriais densas de palavras, que se baseiam na arquitetura de redes neurais artificiais (Mikolov, 2013). As representações distribuídas de palavras preservam algumas propriedades semânticas e sintáticas dos dados textuais aprendidos (Jurafsky & Martin, 2024).

Esta estratégia converte palavras ou sentenças em vetores numéricos, que são compreensíveis para o computador, de forma que boa parte das propriedades sintáticas e semânticas das palavras ou sentenças seja mantida pela sua representação vetorial. Essas representações estão dispostas em um espaço de vetores específico, advindas de uma rede neural artificial.

Cada palavra representa um vetor fixo, sendo chamada de representação vetorial *embedding* estático (Jurafsky & Martin, 2024).

”Dense vectors may also do a better job of capturing synonymy. For example, in a sparse vector representation, dimensions for synonyms like car and automobile dimension are distinct and unrelated; sparse vectors may thus fail to capture the similarity between a word with car as a neighbor and a word with automobile as a neighbor.” Jurafsky & Martin (2024)

Uma forma de entender como o Word2Vec funciona é pensar no contexto entre palavras e em suas adjacências (Mikolov, 2013), em vez de contar com a frequência com que cada palavra ocorre, como fazíamos no BOW. A tarefa torna-se analisar o contexto de cada palavra. O contexto ou janela de contexto é uma sequência de palavras antes e depois de um determinado termo analisado. A intuição final é de que palavras semelhantes ou que possuem relação semântica costumam aparecer na mesma janela de contexto (Bengio & Bengio, 1999; Bengio *et al.*, 2000). Então, podemos utilizar algoritmos para aprender padrões e prever a probabilidade de palavras dado um contexto. Por fim, os padrões ou parâmetros aprendidos pelo algoritmo de predição podem ser usados como representações vetoriais dos termos ou do contexto predito (Jurafsky & Martin, 2024).

A rede neural responsável por representar objetos complexos possui duas formas de realizar essa conversão de representação: CBOW e Skip-Gram.

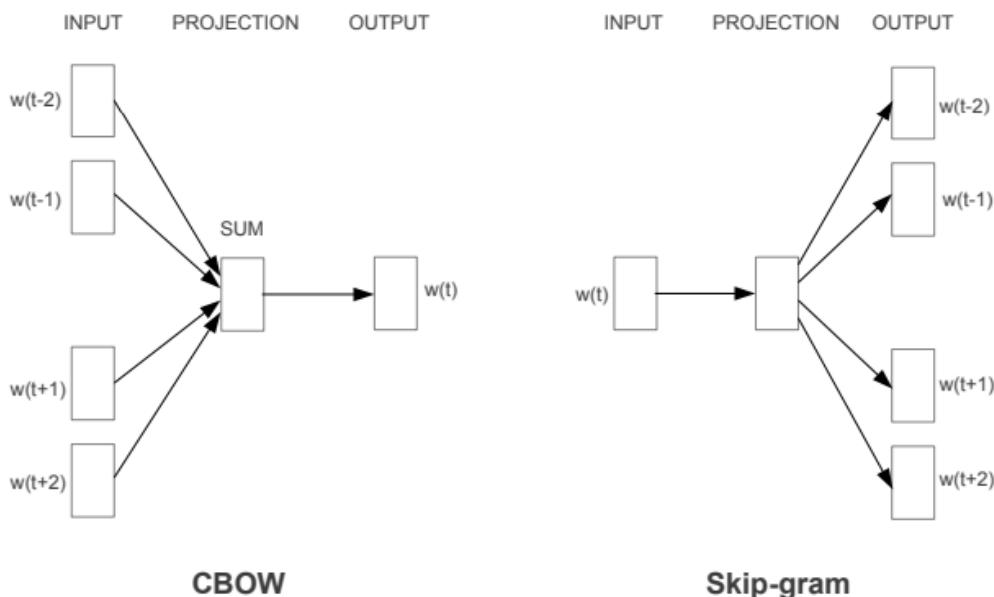


Figura 12: As duas técnicas usadas por Word2Vec, adaptado de Mikolov (2013).

CBOW é um método onde dada uma palavra central e uma determinada quantidade

de termos vizinhos (janela de contexto), as palavras adjacentes são utilizadas para prever o termo do meio. Para isso, uma representação distribuída do contexto é utilizada. Em outras palavras, trata-se da tarefa de tentar adivinhar qual palavra poderia estar no meio, dado um conjunto de palavras vizinhas que fornecem contexto.

Skip-Gram é um método contrário ao que ocorre no *CBOW*, onde a palavra do meio, da janela de contexto, é usada para prever os termos vizinhos. Em outras palavras, o modelo possui uma palavra-alvo e tenta adivinhar quais palavras poderiam ser vizinhas, como se tentasse acertar o contexto com base em uma única palavra. Essa técnica proporciona uma qualidade maior ao resultado dos vetores de palavras. Contudo, também ocorre um aumento na complexidade computacional para processar os dados textuais. *Skip-Gram* apresentou resultados levemente melhores do que *CBOW* (Mikolov, 2013).

3.5.4 Similaridade de Textos

Encontrar a similaridade entre textos é uma tarefa que quantifica o grau de semelhança entre dados textuais (Islam & Inkpen, 2008), que podem ser desde uma palavra, caracteres, documentos até grandes corpora de texto (Gomaa *et al.*, 2013). Ela é amplamente empregada em diversas aplicações de mineração de textos e processamento de linguagem natural, incluindo tarefas como agrupamento de textos (Limiro *et al.*, 2022), classificação de textos (Sebastiani, 2002), recuperação de informações (Pradhan *et al.*, 2015), entre tantas outras tarefas (Chowdhary, 2020).

A ideia subjacente à similaridade de texto é avaliar o quanto próximos ou relacionados os dados textuais são com base em seu conteúdo. Essa relação pode ser estritamente sintática (Gomaa *et al.*, 2013) ou semântica (contextual) (Pradhan *et al.*, 2015). Para isso, utiliza-se uma métrica ou medida de similaridade, como a similaridade de cosseno, a similaridade de Jaccard ou a distância Euclidiana.

Para realizar a tarefa de similaridade de textos, é necessário estruturar os dados textuais em um formato que possibilite a utilização dessas métricas de similaridade, como as estruturas do *bag of words* ou *embeddings*, como vistas anteriormente.

Similaridade de cosseno trata-se de uma métrica que calcula o ângulo entre dois vetores em um espaço vetorial. Quanto mais próximos os vetores, maior a similaridade de cosseno, considerando que os dois vetores são diferentes de zero.

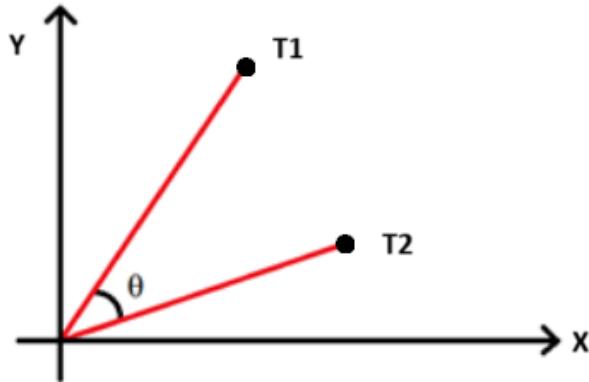


Figura 13: Representação gráfica da similaridade de cossenos.

Para dois vetores T_1 e T_2 , representações vetoriais de um dado textual, a similaridade de cosseno pode ser matematicamente definida como:

$$\text{Similaridade}_{\text{cosseno}}(T_1, T_2) = \cos(\theta) = \frac{T_1 \cdot T_2}{\|T_1\| \|T_2\|} = \frac{\sum_{i=1}^n T_{1,i} T_{2,i}}{\sqrt{\sum T_{1,i}^2} \sqrt{\sum T_{2,i}^2}}$$

A similaridade de cosseno pode assumir valores entre 1 e -1, quanto mais próximos são os vetores analisados, isto é, quanto menor for o ângulo entre eles, maior será o valor de similaridade (Gunawan *et al.*, 2018).

Outras medidas de similaridade existem e podem ser baseadas em informação léxica (string-based ou character-based) ou baseadas em informação semântica (corpus-based ou knowledge-based) (Gomaa *et al.*, 2013; Pradhan *et al.*, 2015; Wang & Dong, 2020).

3.5.5 Agrupamento de Textos

O agrupamento de dados textuais, também referido como clusterização de dados textuais, consiste em agrupar palavras, documentos ou corpora textuais que possuam semelhanças em conjuntos ou grupos (Aggarwal & Zhai, 2012). A ideia central é que os elementos contidos em um grupo são similares entre si, enquanto os elementos de grupos distintos são diferentes (Jain & Dubes, 1988).

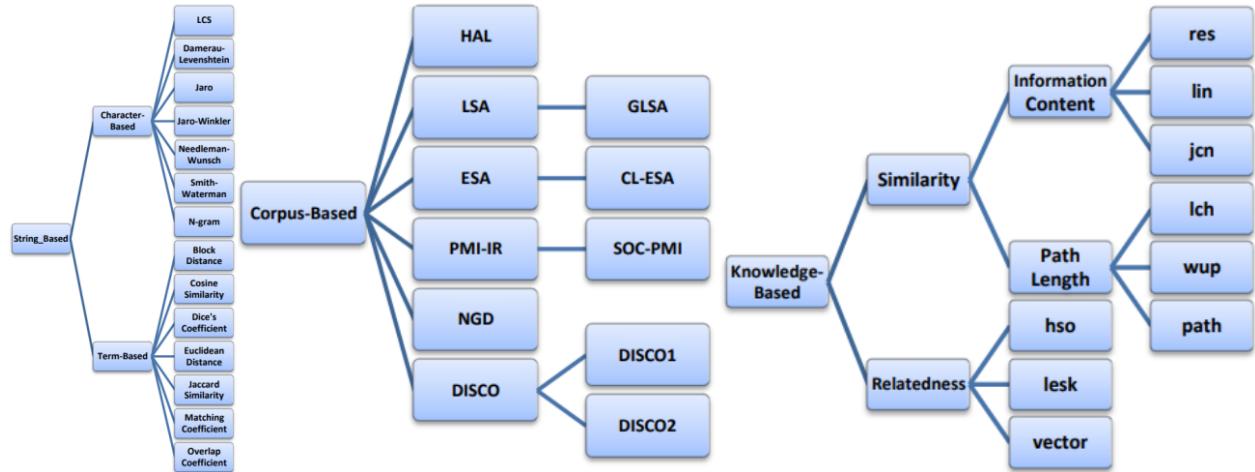


Figura 14: Medidas de similaridade, adaptado de Gomaa *et al.* (2013).

O objetivo do agrupamento de textos é criar agrupamentos de dados textuais que compartilhem características ou conteúdos afins, simplificando assim a organização e a análise de extensos volumes de informações textuais não estruturadas (Aggarwal & Zhai, 2012). Essa metodologia demonstra grande utilidade ao lidar com vastas coleções de documentos, onde se busca identificar padrões, tópicos ou temas subjacentes nos dados (Gomes & Pardo, 2009; Guelpeli, 2012).

Para realizar o agrupamento dos dados textuais, a [Figura 15](#) ilustra o processo em passos de forma simplificada. Um dos primeiros passos é pré-processar os dados textuais, normalizando-os. Por conseguinte, o dado passa por técnicas de vetorização para ser representado em algum espaço de vetores (Gomes & Pardo, 2009), como o BOW ou *embeddings*. Por fim, algum algoritmo de agrupamento, como K-Means (Lloyd, 1982), Agrupamento Hierárquico (Swarndeept Saket & Pandya, 2016) ou DB-Scan (Ester *et al.*, 1996), pode ser usado para obter o rótulo dos grupos encontrados.

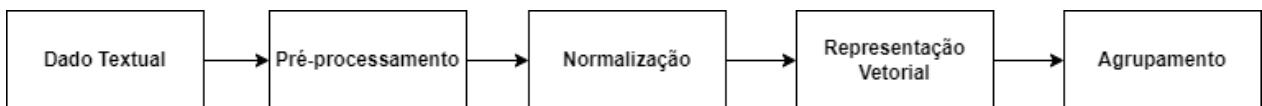


Figura 15: Processo de agrupamento do dado textual por etapas.

3.5.6 Mineração de Tópicos

"Topic models are based upon the idea that documents are mixtures of topics, where a topic is a probability distribution over words." Steyvers & Griffiths (2007)

A modelagem de tópicos, ou mineração de tópicos, é uma técnica que envolve um conjunto

de algoritmos capazes de revelar, descobrir e identificar a estrutura temática em um conjunto de documentos (Steyvers & Griffiths, 2007). Trata-se de uma técnica fundamental em processamento de linguagem natural, pois apoia na sumarização de informações provenientes de dados textuais (Srivastava & Sahami, 2009). O principal objetivo das técnicas de modelagem de tópicos é identificar e extrair temas subjacentes de um conjunto de documentos textuais, permitindo que grandes volumes de textos sejam organizados e representados por tópicos, facilitando a compreensão e a análise de dados (Kherwa & Bansal, 2019; Marcos & Souza, 2019).

Segundo Marcos & Souza (2019):

”A modelagem de tópicos possibilita organizar e resumir, por meio de algoritmos que utilizam métodos estatísticos, conteúdos de arquivos eletrônicos que constituem grandes volumes de dados e informações, chamados de corpus de dados. Sua relevância está no fato de que, em escalas elevadas, torna-se humanamente impossível descobrir e analisar os temas e suas relações a partir de anotações manuais.”

Na modelagem de tópicos, os algoritmos de aprendizagem de máquina costumam utilizar uma abordagem não supervisionada (Churchill & Singh, 2022). Dessa forma, os dados textuais não possuem rótulos históricos, e os resultados não são conhecidos previamente. Contudo, há abordagens supervisionadas (Churchill & Singh, 2022).

Assim como os algoritmos de agrupamento, os algoritmos de mineração de tópicos também necessitam de uma representação numérica para os dados textuais (Kherwa & Bansal, 2019). Representações como BOW ou *embeddings* são utilizadas para a implementação das técnicas de mineração de tópicos (Churchill & Singh, 2022; Mahmood, 2009).

Ao aplicar algoritmos, é possível descobrir automaticamente quais tópicos estão presentes em uma coleção de dados textuais, revelando padrões e relações que não seriam facilmente percebidos pela leitura manual. Os algoritmos mais utilizados na mineração de tópicos são *Latent Dirichlet Allocation* (LDA), *Dynamic Topic Model* (DTM), *Hierarchical LDA* (HLDA), *lda2Vec*, e *BERTopic* (Churchill & Singh, 2022; Kherwa & Bansal, 2019; Mahmood, 2009).

Latent Dirichlet Allocation (LDA) é um algoritmo não supervisionado, não parametrizado e que possui uma abordagem de modelagem probabilística generativa de tópicos (Blei *et al.*, 2003). O LDA assume que cada documento é uma distribuição de probabilidade dos tópicos e que cada tópico é uma distribuição de probabilidade de palavras do documento

(Mahmood, 2009). Normalmente, os dados textuais no LDA são representados como uma matriz de frequência termo-documento, utilizando a estrutura BOW.

A intuição por trás do algoritmo LDA, como ilustrado na [Figura 16](#), consiste em um modelo que utiliza uma abordagem probabilística, ancorada na teoria bayesiana, e parte do princípio de que os documentos contidos em um determinado corpus são representados como uma mistura aleatória de tópicos latentes, e que cada tópico é uma distribuição de palavras que compõe cada um dos documentos textuais (Blei *et al.*, 2003; Marcos & Souza, 2019).

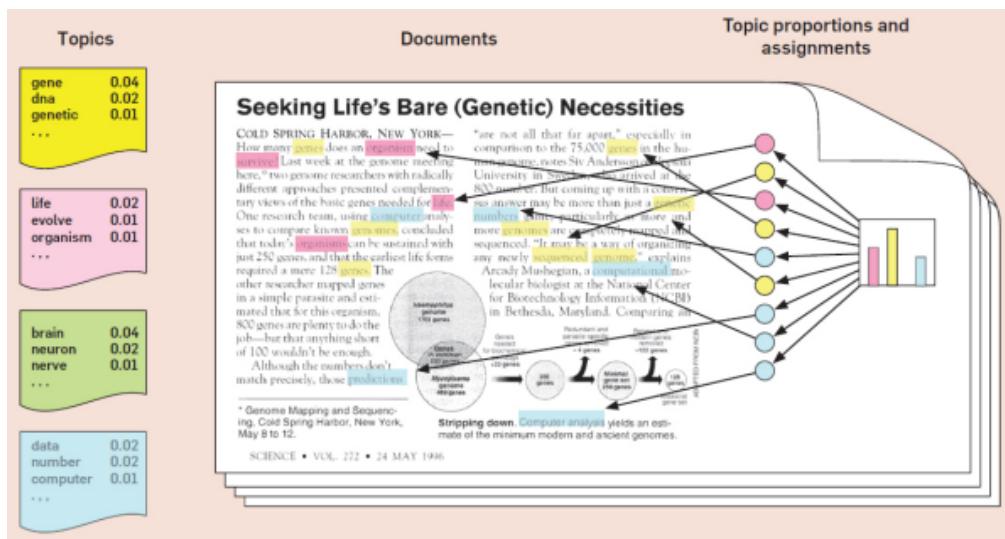


Figura 16: Processo de geração de tópicos pelo LDA, adaptado de Mahmood (2009).

3.6 Inteligência Artificial

"I propose to consider the question, ‘Can machines think?’ This should begin with definitions of the meaning of the terms ‘machine’ and ‘think’." Turing (1950)

Alguns autores definem a Inteligência Artificial (IA) como "O estudo das computações que permitem perceber, raciocinar e agir." (Winston, 1992), ou como "A arte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas." (Smith *et al.*, 2015). Embora seja um campo relativamente novo no conhecimento científico, a inteligência artificial possui inúmeras definições possíveis. Para este projeto, podemos dizer que a inteligência artificial, também conhecida pelo seu acrônimo IA, é um ramo multidisciplinar da ciência da computação (Dwivedi *et al.*, 2021) que se dedica ao desenvolvimento de sistemas computacionais que podem realizar tarefas que normalmente exigem inteligência humana (PK, 1984), como o reconhecimento de padrões, resolução de problemas, tradução automática,

direção com veículos autônomos, reconhecimento de voz, entre outras tarefas, como descreve a Figura 17.

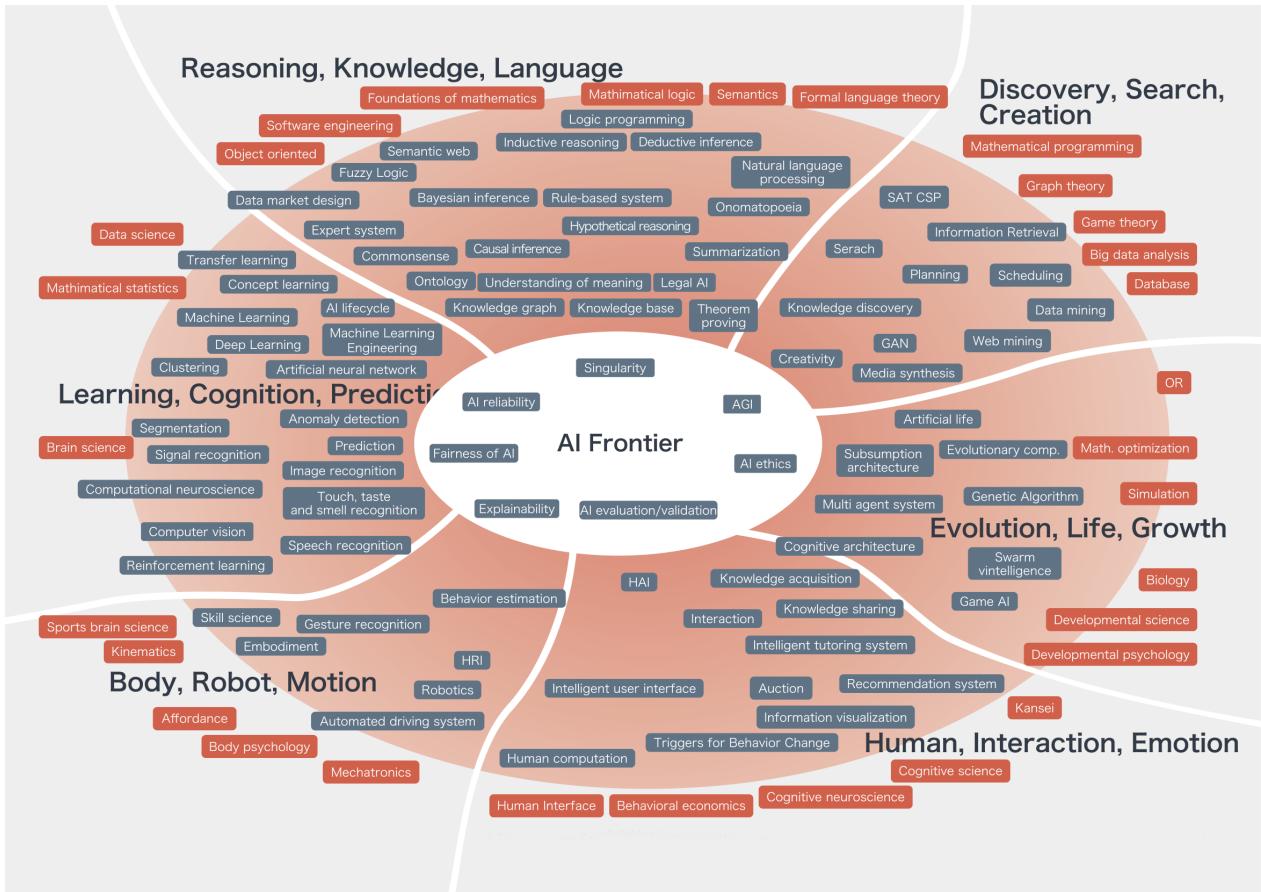


Figura 17: Mapa da fronteira de conhecimento da inteligência artificial JSAI (2021).

3.6.1 Breve História da Inteligência Artificial

Para muitos pesquisadores, a inteligência artificial (IA) teve início como um campo de estudo e pesquisa formal no final da década de 1950, com a publicação de Turing (1950). O termo "inteligência artificial" foi cunhado em 1956 durante a Conferência de Dartmouth, realizada no *Dartmouth College*, nos Estados Unidos (McCarthy *et al.*, 1956).

Para Russell & Norvig (2016) o primeiro trabalho agora reconhecido como de IA foi realizado por Warren McCulloch e Walter Pitts (McCulloch & Pitts, 1943). Os autores se basearam em três fontes: o conhecimento da fisiologia básica e da função dos neurônios no cérebro; uma análise formal da lógica proposicional criada por Russell e Whitehead (Russell & Whitehead, 1910); e a teoria da computação de Turing (1950).

Embora o termo tenha sido usado pela primeira vez nessa época, as ideias e conceitos

que fundamentam a IA remontam a períodos anteriores (Adami, 2021). Ao longo da história, houve várias referências e especulações sobre a possibilidade de máquinas que pudessem exibir inteligência semelhante à humana, como por exemplo (Asimov, 2004).

No entanto, uma conjuntura de eventos proporcionou a formação do campo de pesquisa em IA, que começou a ganhar força na década de 1940, com a criação e o avanço dos computadores digitais (Kaisler, 2016; O'Regan, 2008), o progresso em áreas como a teoria da computação (Turing, 1936), a lógica simbólica (Church, 1936) e as redes neurais artificiais (McCulloch & Pitts, 1943; Rosenblatt, 1958). Dessa forma, a década de 1950 marca o início formal e reconhecido da IA como um campo específico de estudo e pesquisa.

Desde então, a IA tem progredido rapidamente, com avanços em algoritmos, poder computacional, disponibilidade de grandes conjuntos de dados e técnicas de aprendizado de máquina (Brunette *et al.*, 2009; Oke, 2008). Esses avanços impulsionaram a IA para uma ampla gama de aplicações práticas em várias áreas, incluindo visão computacional, processamento de linguagem natural, robótica, jogos, saúde, finanças e muito mais (Toosi *et al.*, 2021).

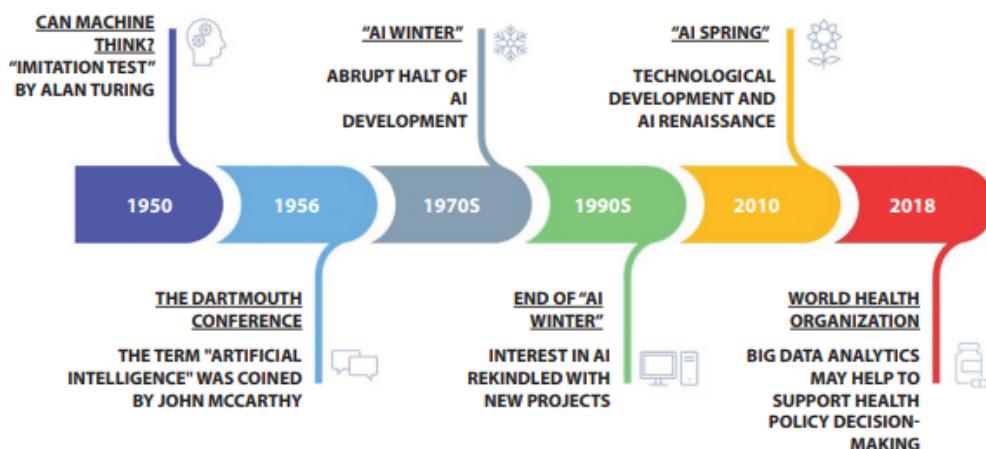


Figura 18: Diagrama de linha do tempo mostrando a história da inteligência artificial adaptado de Bellini *et al.* (2022).

3.6.2 Campos da Inteligência Artificial

Atualmente, o campo da IA está dividido em várias subáreas com abordagens distintas, como a visão computacional, agentes inteligentes, IA explicável (*explainable AI*), sistemas de recomendação, sistemas inteligentes, reconhecimento de padrões, mineração de dados, processamento de linguagem natural, aprendizado de máquina, aprendizado profundo, entre outros tópicos Figura 19.



Figura 19: Campos da IA segundo JSAI, 2021.

O campo da inteligência artificial está em franca expansão, impulsionado pelos avanços tecnológicos e melhorias em algoritmos de aprendizado de máquina. Essa evolução tem levado à adoção crescente da IA em diversos setores, incluindo saúde (Yu *et al.*, 2018), finanças (Enholm *et al.*, 2022), logística (Woschank *et al.*, 2020), marketing (Chintalapati & Pandey, 2022), entre muitos outros setores, as soluções buscam otimizar processos e oferecer soluções para os problemas da sociedade.

3.7 Aprendizado de Máquina

O aprendizado de máquina é uma área de estudo da inteligência artificial que se concentra no desenvolvimento de algoritmos capazes de aprender padrões a partir de dados históricos (Hastie *et al.*, 2009). Ao invés de serem explicitamente programados para realizar uma tarefa específica, como na programação estruturada determinística, os sistemas de aprendizado de máquina conseguem aprender a partir de exemplos e dados (Andrew, 2016), identificando padrões, correlações e relações nos dados para fazer previsões ou tomar decisões.

Uma definição mais formal, parafraseando Tom Michell, é que o aprendizado de máquina trata-se de um programa de computador que aprende a partir de uma experiência E em relação a uma tarefa T , e cujo critério de aprendizado é medido por uma medida de desempenho P . Espera-se que, à medida que este programa de computador execute T , o seu desempenho P melhore com o conjunto de experiência E (Mitchell *et al.*, 1997). A definição de Michell traz a noção de que o aprendizado é concebido por meio de experiências pré-estabelecidas, já conhecidas, em uma base de conhecimento acerca de uma tarefa à que um programa de computador executa.

Para realizar essa tarefa, os algoritmos de aprendizado de máquina utilizam algoritmos matemáticos para identificar padrões e relações nos dados de treinamento (James *et al.*, 2013). Esses algoritmos podem ser aplicados em diversas áreas, como, por exemplo, reconhecimento de fala, reconhecimento de imagem, análise de dados, modelos preditivos, modelos prescritivos e sistemas de recomendação (Shinde & Shah, 2018).

3.7.1 Aprendizado Supervisionado

Aprendizado supervisionado é uma das técnicas de aprendizado de máquina nas quais um modelo é treinado utilizando exemplos rotulados de dados (Domingos, 2015). Esses exemplos são compostos por pares de entrada e saída desejada, onde a saída desejada é conhecida e fornecida durante o processo de treinamento com a entrada (Bishop & Nasrabadi, 2006).

O objetivo do aprendizado supervisionado é aprender a função, ou modelo, que mapela as entradas (dados de treino) para as saídas desejadas (rótulos dos dados de treino) (Bishop & Nasrabadi, 2006), de modo que o modelo possa fazer previsões para novos dados

de entrada desconhecidos (dados de teste). O diagrama da [Figura 20](#) descreve esse processo. Durante o treinamento, o modelo faz previsões para as entradas de treinamento, sendo ajustado para minimizar a diferença entre as previsões e as saídas desejadas. Essa diferença é medida por uma função de perda, função de ajuste ou função de otimização ([Hastie et al.](#), 2009), também conhecidas como medidas de avaliação de aprendizado supervisionado.

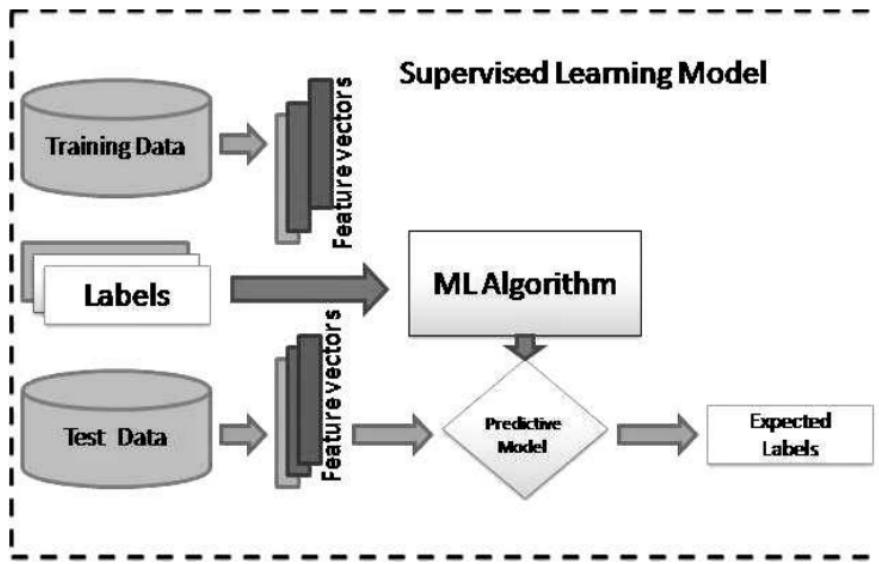


Figura 20: Um diagrama esquemático que descreve o modelo de aprendizado supervisionado, adaptado de [Muhammad & Yan \(2015\)](#).

As tarefas de aprendizado supervisionado incluem classificação e regressão ([Hastie et al.](#), 2009; [James et al.](#), 2013).

3.7.2 Redes Neurais Artificiais

Uma rede neural artificial é uma representação matemática inspirada no cérebro humano que abstrai, de forma simplificada, o funcionamento de um cérebro, modelando a arquitetura de redes neurais em estruturas computacionais descritas em modelos matemáticos calculáveis ([Haykin, 2001](#)). Neurônios artificiais e redes neurais artificiais podem ser usados com algoritmos de *machine learning* supervisionados ([Muhammad & Yan, 2015](#)).

O campo de redes neurais artificiais teve seu início com o neurônio artificial de McCulloch-Pitts ([McCulloch & Pitts, 1943](#)). Um diagrama representando um neurônio artificial pode ser visto na [Figura 21](#).

Um neurônio artificial $N(\cdot)$ é composto por sinais de entrada x_1, x_2, \dots, x_m que são da-

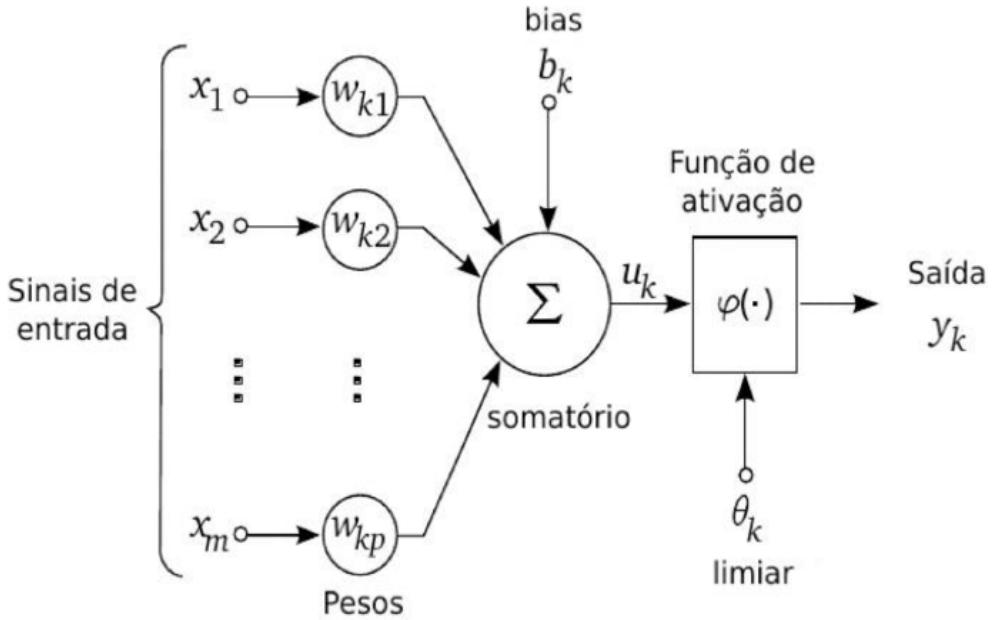


Figura 21: Uma representação do neurônio artificial de McCulloch-Pitts, adaptado de Haykin (2001); McCulloch & Pitts (1943).

dos ou informações que possuem pesos distintos $w_{k1}, w_{k2}, \dots, w_{km}$, denotando a importância de cada entrada. Após a ponderação da entrada $x_i w_{ki}$, os sinais passam por uma função de agregação $\sum_{i=1}^m x_i w_{ki} + b_k$ que soma os valores e as importâncias das entradas em um único valor. Por fim, este valor agregado serve de entrada para uma função específica $\varphi(\cdot)$, chamada de função de ativação. Esta função possui características específicas dependendo da sua expressão analítica, e é incorporada em cada neurônio artificial. Em especial, possui um limiar de ativação que representa um valor dentro de seu domínio a partir do qual a função retorna um valor específico que é exposto em sua saída y_k .

$$N(b_k, x_1, x_2, \dots, x_m) = \varphi\left(\sum_{i=1}^m x_i w_{ki} + b_k\right) = y_k$$

A rede neural artificial é composta por camadas de neurônios artificiais conectados entre si (Popescu *et al.*, 2009). Na Figura 22, pode-se ver um diagrama representativo de uma rede neural artificial com múltiplos neurônios artificiais do tipo *perceptrons* (MLP). Numa MLP, cada camada possui seus neurônios artificiais que incorporam informações, processam-nas e retornam para a próxima camada de neurônios artificiais em um fluxo contínuo e unidirecional (Popescu *et al.*, 2009). Em outras palavras, a primeira camada recebe a informação em cada entrada dos neurônios artificiais. Para cada neurônio artificial, a camada processa e repassa

as saídas que, por sua vez, são captadas pelas entradas dos neurônios artificiais da camada seguinte, e o fluxo de dados continua até a camada de saída final.

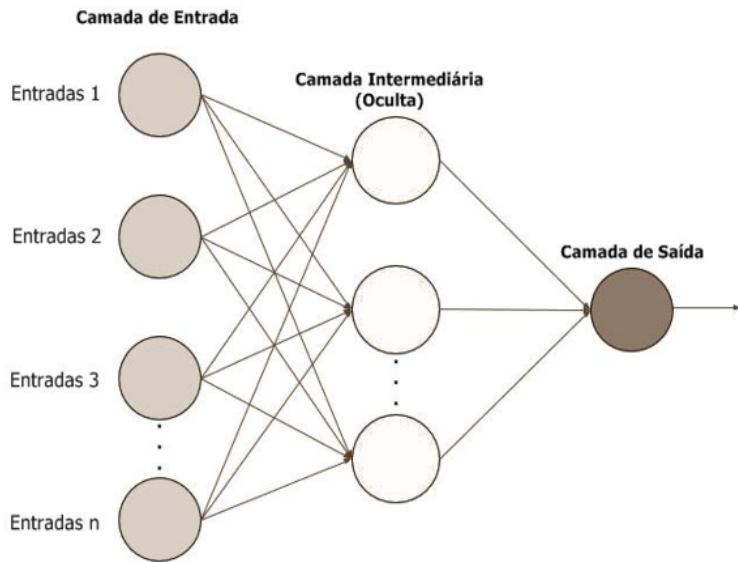


Figura 22: Uma representação da rede neural multi-perceptron de alimentação direta (*feed-forward*), adaptado de Popescu *et al.* (2009).

Os pesos ou parâmetros associados a cada neurônio artificial contém o aprendizado da rede, capturando os padrões dos dados a serem aprendidos. A atualização dos pesos na rede neural artificial passa por um processo iterativo chamado backpropagation (Rojas & Rojas, 1996). Neste processo de backpropagation, os pesos são gradualmente corrigidos a cada iteração de entrada dos dados, para incrementar os aprendizados da rede neural artificial até atingir algum critério de parada (Jurafsky & Martin, 2024, c.7-7.5).

As redes neurais artificiais possuem diversas estruturas de funcionamento, variando suas funções de ativação (Rasamolena *et al.*, 2020), topologias e arquiteturas de ligação entre seus neurônios artificiais (Ibnu *et al.*, 2019; Vogels *et al.*, 2005; Wilamowski, 2009).

3.7.3 Aprendizado Não-Supervisionado

“*Unsupervised learning or learning without a teacher.*” (Hastie *et al.*, 2009, c.14)

O aprendizado não supervisionado é uma das técnicas de aprendizado de máquina que se concentra em analisar dados não rotulados para descobrir padrões latentes (Hastie *et al.*, 2009). Em contraste com o aprendizado supervisionado, que requer dados rotulados, o aprendizado não-supervisionado não depende de informações de saída conhecidas (Faceli *et al.*,

2011). O objetivo do aprendizado não-supervisionado é permitir que o algoritmo explore os dados por conta própria e descubra relações e padrões ocultos nos dados (Domingos, 2015).

Segundo Ghahramani (2003):

"In unsupervised learning the machine simply receives inputs x_1, x_2, \dots , but obtains neither supervised target outputs, nor rewards from its environment. It may seem somewhat mysterious to imagine what the machine could possibly learn given that it doesn't get any feedback from its environment".

Como não há informações sobre o que aprender nos dados usando técnicas não-supervisionadas, os algoritmos passam a ser desenvolvidos e utilizados com finalidades muito específicas e bem direcionadas, como na identificação de agrupamentos em dados, redução de dimensionalidade de dados (compactar dados), mineração de regras de associação, detecção de anomalias, entre outras aplicações (Dike *et al.*, 2018; Ghahramani, 2003; Hastie *et al.*, 2009).

3.7.4 Agrupamento

Trata-se de uma técnica de aprendizado não-supervisionado usada para separar um conjunto de dados em grupos, agrupamentos ou *clusters* com base em suas características e similaridades. Essa técnica visa identificar padrões e estruturas ocultas nos dados (Faceli *et al.*, 2011).

Como o agrupamento é uma técnica de aprendizado não-supervisionado, não requer rótulos de saída para treinar o algoritmo. Ao invés disso, o algoritmo utiliza propriedades como distância, proximidade, densidade ou similaridade para formar os agrupamentos (Faceli *et al.*, 2011). O objetivo de agrupar é que objetos que pertençam a um mesmo grupo possuam características semelhantes, enquanto que os objetos que estão em grupos distintos não são similares (Hastie *et al.*, 2009).

Alguns dos algoritmos mais conhecidos de agrupamento são: *K-Means*, *K-Medoids*, *Bisect K-Means*, Agrupamento Hierárquico, *DBScan*, *Birch*, *Gaussian Mixture Models* (GMM), entre outros algoritmos (Xu & Tian, 2015).

K-means é uma das técnicas de agrupamento mais conhecidas e utilizadas para gerar grupos em um conjunto de dados (Hastie *et al.*, 2009, c.14). O *K-Means* usa medidas de

distância como função de similaridade (Hastie *et al.*, 2009; Kapil & Chawla, 2016, CH:14). O *K-Means* é baseado em partições e, para cada partição, divide os dados em regiões bem delimitadas em um espaço de vetores, onde cada objeto ou instância pertence somente a uma única partição (Xu & Tian, 2015).

O algoritmo começa com a escolha de k elementos representativos de um grupo (David & Sergei, 2006), chamados de centróides iniciais, onde k é o número de *clusters* desejados e previamente definidos (Cui *et al.*, 2020). Em seguida, o algoritmo atribui todos os pontos de dados ao centróide mais próximo com base na distância entre eles (MacQueen, 1967). Após a atribuição, os centroides são recalculados como a média dos pontos em cada *cluster*, assumindo novas posições atualizadas (MacQueen, 1967). Essas etapas de atribuição e atualização de centroides são repetidas iterativamente até que as partições estejam estabilizadas de acordo com algum critério de parada (Hastie *et al.*, 2009, c.14).

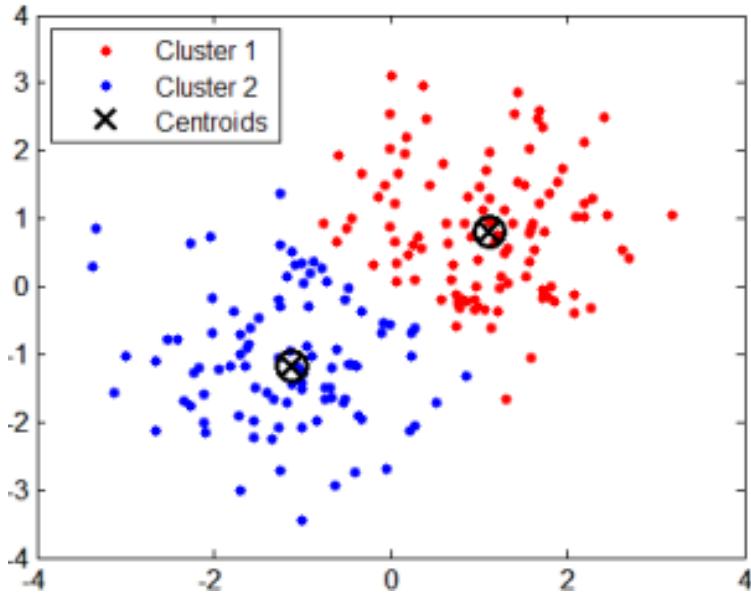


Figura 23: Uma representação em duas dimensões de uma agrupamento de dados utilizando o algoritmo *K-means*, adaptado de Burkardt (2009).

O resultado final é uma divisão dos dados em grupos, cada um representado por um centróide, como pode ser visto em Figura 23. Embora seja eficiente, a escolha inicial dos centróides e o valor de k podem afetar os resultados, e é comum executar o algoritmo várias vezes para obter a melhor solução (David & Sergei, 2006).

A inércia, no contexto do algoritmo K-Means, é uma medida que quantifica o quanto coeso é um grupo em relação ao seu centróide. Esta medida é usada para avaliar a qualidade dos

grupos formados durante o processo de agrupamento, isto é, o quanto os pontos de dados estão dispersos dentro de cada grupo (medida *intra-cluster*) (Faceli *et al.*, 2011). A inércia é calculada somando as distâncias quadradas de cada ponto de dados para o centróide do grupo ao qual pertence e, em seguida, somando essas distâncias para todos os pontos de dados e grupos (Faceli *et al.*, 2011).

Além da inércia, existem outras medidas para avaliar a qualidade dos grupos encontrados pelos algoritmos de agrupamento, como por exemplo o índice Davies-Bouldin, o Score de Silhueta, o Score de Silhueta Simplificada, o índice Calinski-Harabaz, entre outras medidas (Wegmann *et al.*, 2021).

3.7.5 Redução de Dimensionalidade

Um dos problemas enfrentados pelos algoritmos de agrupamento é a "maldição da dimensionalidade" (Donoho *et al.*, 2000, c.7). Esse fenômeno decorre da alta dimensionalidade apresentada pelos dados que, por sua vez, podem conter um grande número de dimensões.

Quando se trabalha com dados não-estruturados, sua representação costuma ser em estruturas de alta-dimensionalidade, como dados textuais, áudios e imagens. Em um cenário de alta-dimensionalidade, a visualização dos dados é inviável. Nesse contexto, muitas dimensões podem ser irrelevantes, os dados tornam-se esparsos, com menor densidade, e as medidas de distância entre objetos passam a ser incoerentes (Wegmann *et al.*, 2021).

Neste sentido, a redução de dimensionalidade pode ser realizada por diversas técnicas, como a seleção de dimensões (Venkatesh & Anuradha, 2019), engenharia de recursos (Dong & Liu, 2018) ou por algoritmos de redução de dimensionalidade, como a análise de componentes principais (PCA), análise do discriminante linear (LDA), análise fatorial (FA), análise de componentes independentes (ICA), decomposição em valores singulares (SVD), mapeamento isométrico (ISOMAP), incorporação de vizinho estocástico distribuído (t-SNE), entre outros (Sorzano *et al.*, 2014).

Com estes algoritmos de redução de dimensionalidade, torna-se possível mitigar os efeitos da "maldição da dimensionalidade" ao custo de perda de informação no processo (Sorzano *et al.*, 2014). Contudo, a redução possibilita a visualização dos dados, a remoção de

dados irrelevantes e a manipulação eficiente dos dados. Alguns desses algoritmos de redução de dimensionalidade são utilizados em conjunto com a representação de dados textuais *bag of words* (BOW).

3.7.6 Arquitetura *Transformers*

Os *transformers* são um tipo de rede neural artificial não-recorrente com uma arquitetura específica que utiliza como base uma técnica chamada mecanismo de atenção (auto-atenção) ou atenção multi-cabeças (*multi-head attention*) (Vaswani *et al.*, 2023). O *transformer* é a arquitetura padrão para a construção de grandes modelos de linguagem, do inglês *Large Language Models* (LLM) (Jurafsky & Martin, 2024, c.9), os modelos populares BERT (*Bi-directional Encoder Representations from Transformers*) (Devlin, 2018) e GPT (*Generative Pre-trained Transformer*) (Radford & Narasimhan, 2018) são construídos utilizando esta arquitetura.

A arquitetura de redes neurais artificiais dos *transformers* é utilizada para processamento de dados sequenciais, como os dados textuais (Sutskever, 2014), áudios, imagens e vídeos. Eles são amplamente empregados em diversas tarefas no campo da IA (Lin *et al.*, 2022), como a tradução automática de textos (Sutskever, 2014), processamento de áudio (Latif *et al.*, 2023), visão computacional (Khan *et al.*, 2022), séries temporais (Wen *et al.*, 2022), aprendizado por reforço (Li *et al.*, 2023), e detecção de objetos (Carion *et al.*, 2020). São aplicados também em outras disciplinas, como nas áreas da saúde (Nerella *et al.*, 2023), farmacologia (Jiang *et al.*, 2024) e química (Sultan *et al.*, 2024).

Os *transformers* possibilitam a captura de contexto, estabelecendo conexões latentes entre os dados de entrada e a capacidade de memorizar essas conexões para realizar alguma tarefa que exija contexto e memória (Khan *et al.*, 2022; Lin *et al.*, 2022).

Por exemplo, na tarefa de tradução automática de textos, sabemos que o texto possui uma sequência de geração e que as palavras próximas entre si têm o mesmo contexto. Na frase em inglês: ”*Queen was a British rock band, founded in 1970 and active, under its classical formation, until 1991. The band's music is also known for being highly eclectic, varying between various aspects of rock.*”; no exemplo, a palavra ”*band*”, na segunda frase,

refere-se à "Queen". Para seres humanos, ao ler a frase, a relação entre as palavras é direta, e associamos uma palavra à outra. Contudo, para a máquina, essa relação não é trivial e precisa ser estabelecida.

O mecanismo de atenção permite que o modelo *transformer* consiga analisar diferentes partes da entrada dos dados ao mesmo tempo, ajustando dinamicamente a relevância de cada componente da entrada em relação às outras partes (Vaswani *et al.*, 2023). Isso permite que o modelo capture diferentes tipos de relações contextuais entre as entradas, aumentando a riqueza das representações (Lin *et al.*, 2022).

Ao contrário das redes neurais artificiais recorrentes (RNN), a arquitetura de rede neural artificial precursora dos *transformers*, que processam sequências de forma sequencial, os *transformers* processam toda a sequência de entrada simultaneamente (Vaswani *et al.*, 2023). Isso resulta em um treinamento mais rápido e eficiente, especialmente em grandes conjuntos de dados (Vaswani *et al.*, 2023).

O funcionamento da arquitetura de redes neurais *transformer* foi formalmente definido e pode ser aprofundado por meio dos trabalhos de Lin *et al.* (2022); Vaswani *et al.* (2023).

3.7.7 Grandes Modelos de Linguagem (ou *Large Language Models* - LLMs)

"LLMs are large-scale, pre-trained, statistical language models based on neural networks." Minaee *et al.* (2024)

A linguagem é uma habilidade que os humanos começam a desenvolver e utilizar para comunicação e expressão desde o início da vida (Pinker, 2003). As máquinas, por outro lado, não possuem essa habilidade de linguagem, e temos que elaborar heurísticas para que elas possam adquirir um comportamento que se aproxime dessa habilidade humana (Minaee *et al.*, 2024; Zhao *et al.*, 2023).

A abordagem de modelos de linguagem é a mais avançada para conceder às máquinas a capacidade de compreensão de textos e, por consequência, a compreensão da linguagem (Zhao *et al.*, 2023). O estudo dos modelos de linguagem vem de longa data, desde o **modelo de linguagem estatístico** mais simples, os **n-grams**, criados na década de 1990, que utilizava um modelo preditivo baseado na suposição de Markov. A ideia básica dos modelos de linguagem estatísticos é de predizer a próxima palavra, com base na frequência das palavras

mais recentes, dado um contexto (Jurafsky & Martin, 2024, c.3).

A evolução dos modelos passa pelos ***modelos de linguagem neural (NLM)***, que utilizam arquiteturas de redes neurais artificiais simples para uma representação distribuída das palavras (Bengio *et al.*, 2000; Mikolov, 2013; Mikolov *et al.*, 2013). Desta forma, a predição de uma próxima palavra é condicionada ao contexto latente agregado pela rede neural artificial.

Os NLMs permitiram o surgimento dos ***modelos de linguagem pré-treinados (PLM)*** com arquiteturas de redes neurais artificiais recorrentes e uma maior capacidade de representação do contexto das palavras, habilitando a capacidade de pré-treinamento, ajuste das representações para tarefas específicas e transferência de aprendizado entre máquinas (Zhao *et al.*, 2023). Por fim, os ***grandes modelos de linguagem (LLM)*** evoluíram a capacidade dos PLMs com a possibilidade de paralelização do treinamento, utilizando mecanismos de auto-atenção com a arquitetura de rede neural artificial *transformer* (Vaswani *et al.*, 2023), aumentando vertiginosamente a escala de dados processados e os parâmetros no modelo de linguagem, chegando a ordem de bilhões de parâmetros (Minaee *et al.*, 2024; Zhao *et al.*, 2023). Um exemplo de LLM é o GPT-3, com aproximadamente 175 bilhões de parâmetros, treinado com cerca de um trilhão de *tokens* (Brown, 2020).

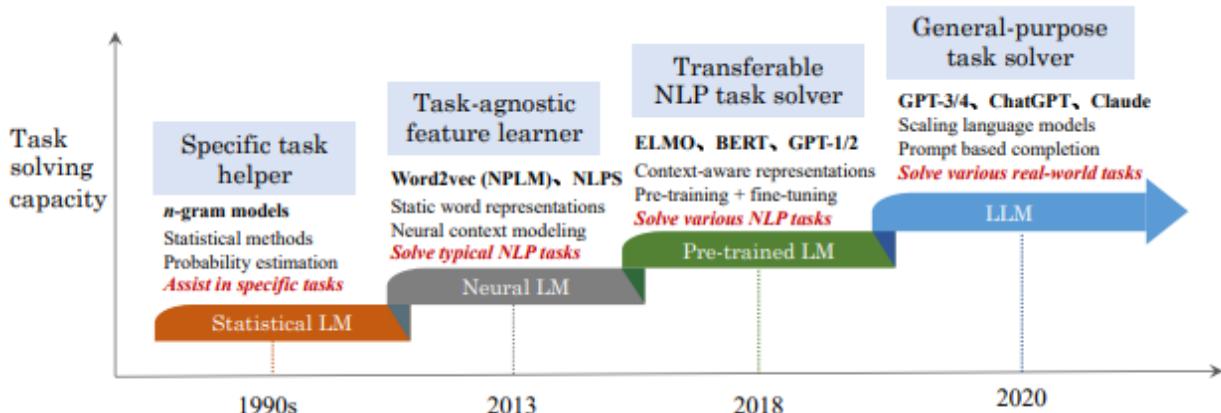


Figura 24: O processo de evolução das gerações de modelos de linguagem de acordo com Zhao *et al.* (2023).

Segundo Jurafsky & Martin (2024):

"Large language models are mainly trained on text scraped from the web, augmented by more carefully curated data. Because these training corpora are so large, they are likely to contain many natural examples that can be helpful for NLP tasks, such as question and answer pairs (for example from FAQ lists), translations of sentences between various languages, documents together with their summaries, and so on."

Os LLMs apresentam uma forte capacidade de linguagem, seja para entendimento ou geração automática de texto. Além disso, apresentam habilidades emergentes que não estão presentes em outros tipos de modelos de linguagem, como a capacidade de aprender novas tarefas, dado um conjunto de exemplos, uma base de conhecimento ou instruções, em tempo de execução (Minaee *et al.*, 2024).

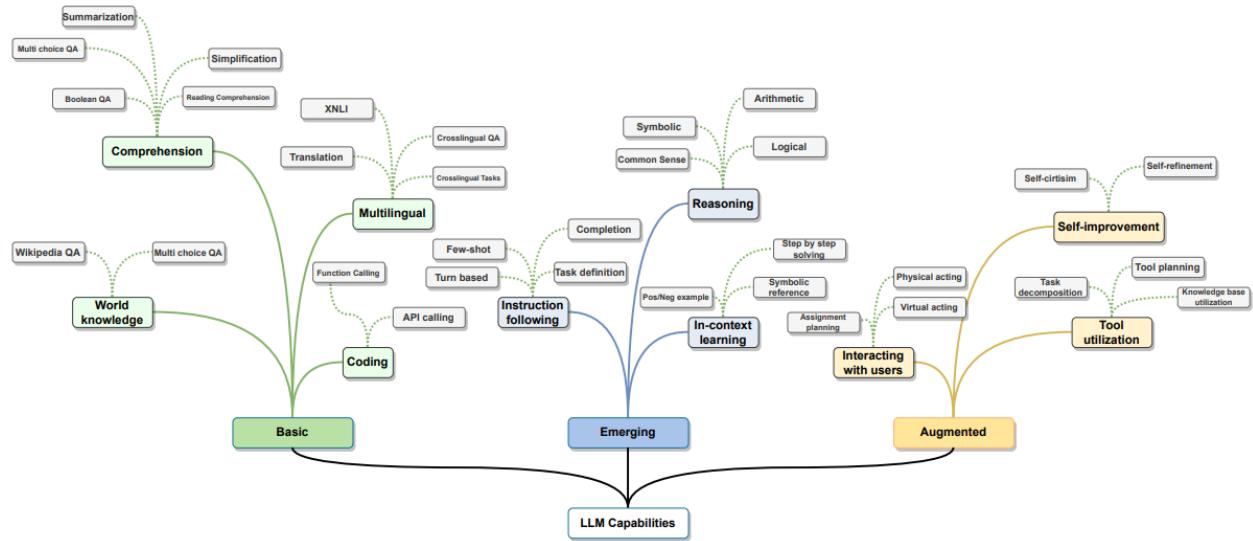


Figura 25: O processo de evolução das gerações de modelos de linguagem de acordo com Zhao *et al.* (2023).

Uma das capacidades dos LLMs é a possibilidade de aumentar seu conhecimento utilizando uma base de conhecimento para um determinado domínio, permitindo o uso de uma fonte de conhecimento externa ao seu treinamento para interação com usuários ou o ambiente (Asai *et al.*, 2021; Mialon *et al.*, 2023).

A capacidade dos LLMs de utilizar uma base de conhecimento externa e a interação com usuários e o ambiente será utilizada neste projeto.

Os métodos de quantização de parâmetros são utilizados para gerar grandes modelos de linguagem quantizados que, por sua vez, são versões de LLMs otimizadas para reduzir o uso de recursos computacionais e a quantidade de memória necessária, mantendo um desempenho aceitável. A quantização é uma técnica que envolve a redução da precisão dos números que representam os parâmetros do LLM, como os pesos das *Recurrent Neural Network* RNNs e as ativações, geralmente trocando o tipo de dados de ponto flutuante (números reais) por inteiros (números inteiros). A quantização resulta em modelos que ocupam menos armazenamento, são mais rápidos e eficientes, facilitando sua implementação em dispositivos com recursos

limitados, como máquinas locais, smartphones e dispositivos *IoT* (Zhao *et al.*, 2023).

Dado o contexto de pesquisa, será utilizado LLMs quantizados.

3.7.8 Geração Aumentada via Recuperação (RAG)

Embora os LLMs possuam muitas capacidades, também apresentam alguns desafios na sua utilização, como as alucinações (Marcus, 2020), imprecisões na geração de texto, o custo elevado de retreinamento ou ajuste fino, e a geração de texto muito genérica ou com conteúdo irrelevante para uma determinada tarefa (Gao *et al.*, 2023).

Uma forma de lidar com alguns desses problemas é fornecer ao LLM uma base de conhecimento externa para consulta na execução de tarefas. Para este processo, dá-se o nome de geração aumentada via recuperação (Lewis *et al.*, 2020). O mecanismo RAG muito utilizada na elaboração de agentes conversacionais (Zhao *et al.*, 2023, c.9.6).

Segundo Gao *et al.* (2023):

”Retrieval-Augmented Generation (RAG) enhances LLMs by retrieving relevant document chunks from external knowledge base through semantic similarity calculation. By referencing external knowledge, RAG effectively reduces the problem of generating factually incorrect content. Its integration into LLMs has resulted in widespread adoption, establishing RAG as a key technology in advancing chatbots and enhancing the suitability of LLMs for real-world applications.”

Uma base de conhecimento pode ser um conjunto de documentos, ou seja, um conjunto de dados textuais, que passa pelo seguinte processo em sua construção e uso: os documentos são carregados e pré-processados; após isso, passam por uma etapa de vetorização latente (*embeddings*), indexação em um banco de dados de vetores; a seguir, os documentos vetorializados mais relevantes para a tarefa são selecionados de acordo com a entrada do usuário e incorporados nas instruções para a execução da tarefa junto ao LLM. A Figura 26 exemplifica o processo de incorporação de memória ao LLM.

Utilizar o mecanismo RAG torna o poder de execução de tarefas pelo LLM muito mais acurado, pois o LLM tem maior contexto e informações específicas para formular uma resposta ao usuário (Mialon *et al.*, 2023).

Há diferentes formas de aplicar RAG à uma LLM, incluindo: RAG ingênuo, RAG avançado e RAG modular. A Figura 27 traz uma comparação entre as formas de aplicação de RAG.

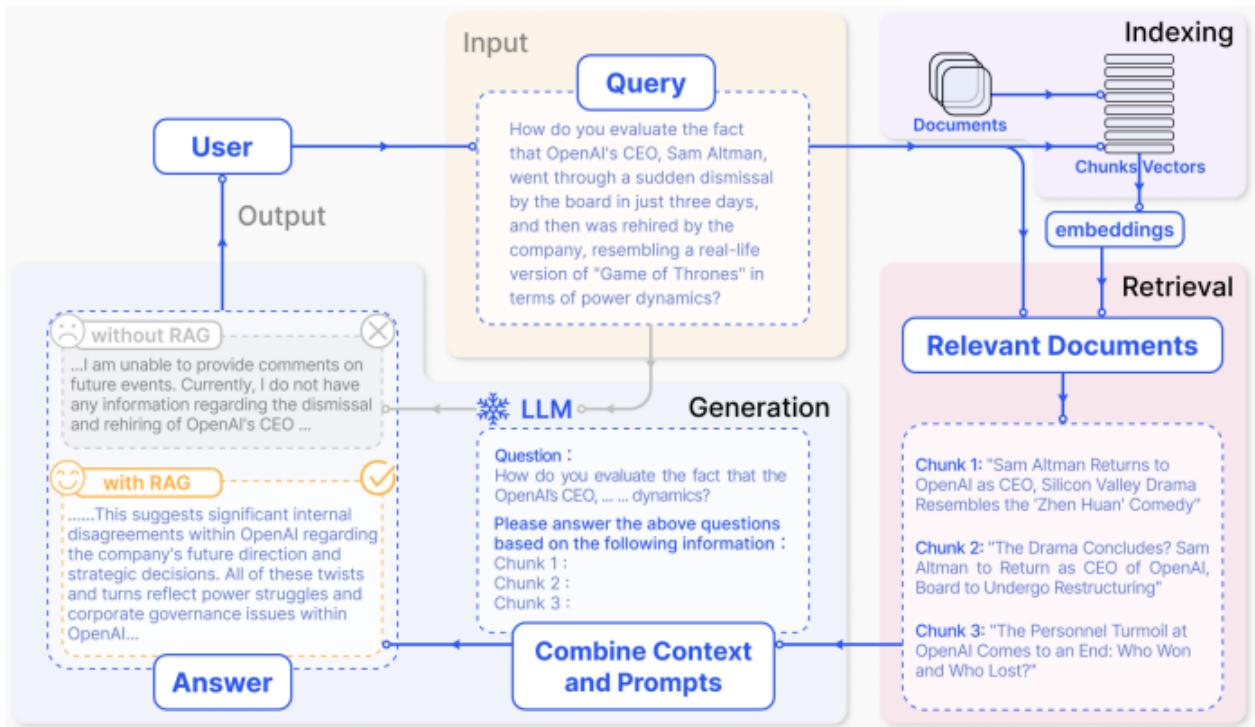


Figura 26: O processo de RAG aplicado num contexto de perguntas e respostas de acordo com Gao *et al.* (2023).

Para este trabalho, vamos utilizar a aplicação do mecanismo RAG ingênuo.

3.7.9 Outras Categorias de Aprendizado

Além do aprendizado supervisionado e do aprendizado não-supervisionado, existem outros tipos de aprendizado de máquina que podem ser usados para a resolução de problemas, como o aprendizado por reforço (Wang *et al.*, 2022), aprendizado por reforço com retorno humano (Kaufmann *et al.*, 2023), aprendizado semi-supervisionado (Van Engelen & Hoos, 2020), aprendizado auto-supervisionado (Gui *et al.*, 2024; Schiappa *et al.*, 2023) e aprendizado por transferência (Weiss *et al.*, 2016; Zhuang *et al.*, 2020).

As categorias de aprendizado podem ser utilizadas de forma isolada ou de maneira conjunta no aprendizado de máquina para alguma tarefa. Por exemplo, quando trabalhamos com grandes modelos de linguagem, podemos ter o emprego de aprendizado auto-supervisionado, supervisionado, não-supervisionado, aprendizado por transferência e aprendizado por reforço com retorno humano (Raiaan *et al.*, 2024).

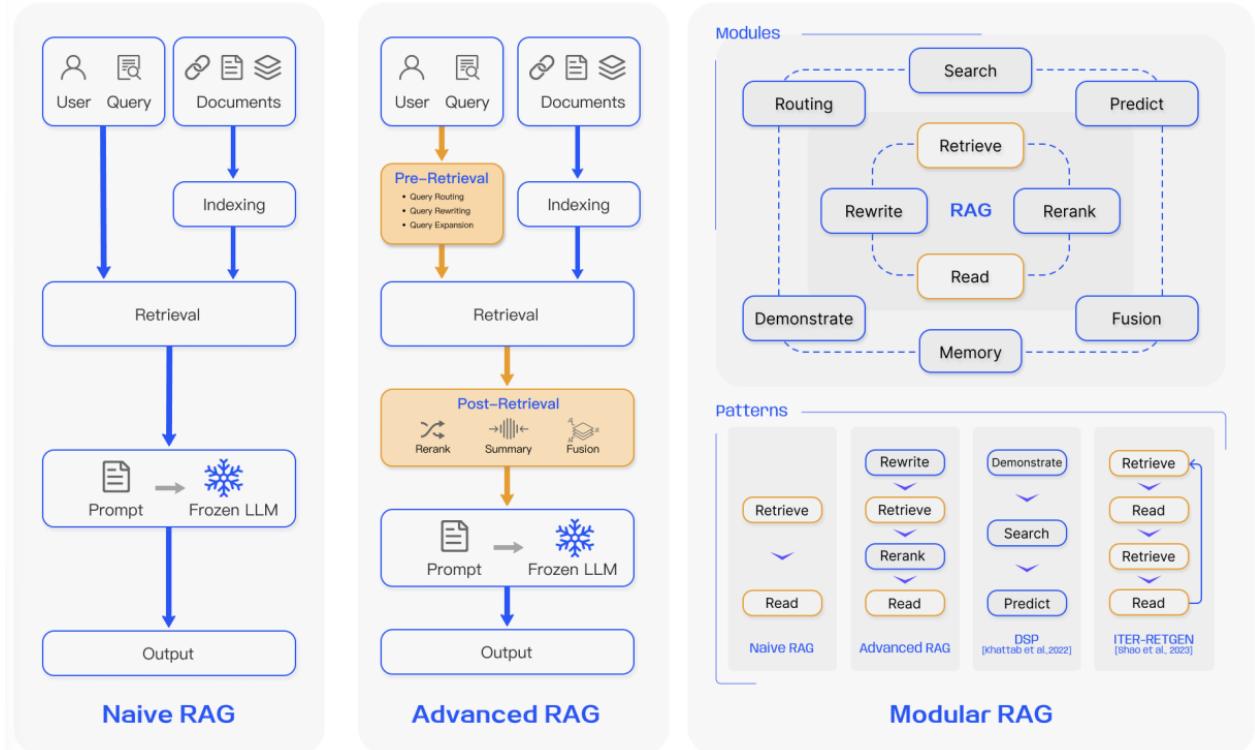


Figura 27: Comparação entre os três paradigmas do mecanismo RAG de acordo com Gao *et al.* (2023).

3.8 Visualização de Dados

Uma visualização trata-se de uma representação visual interativa que favorece a análise de dados por meio de um diagrama, formando uma imagem sobre algo com o intuito de fornecer a cognição humana e a capacidade de perceber e processar informações (Van Wijk, 2005).

O processo de criação de visualização de dados passa por diversas etapas até a construção de uma visualização de dados que seja útil para aqueles que a consomem. Para chegar a uma visualização de dados, é preciso passar por um processo de entendimento dos dados: seja por sua semântica, estrutura e características, como tipos, tamanhos e formatos; passar por um processo de análise exploratória, interpretação e, por fim, pela comunicação (Qin *et al.*, 2020).

As visualizações de dados são desenvolvidas com o intuito de facilitar a análise e a compreensão de um grande volume de dados, tornar evidentes informações latentes, promover uma melhor interpretação de informações complexas e comunicar informações de forma eficiente (Chan, 2006). Uma visualização de dados tem a capacidade de impulsionar o sistema cognitivo do ser humano para analisar informações de forma mais eficiente e eficaz, sumarizando

a informação (Khan & Khan, 2011).

Neste trabalho, toda os dados coletados são processados e se tornarão visualizações de dados interativas ou ferramentas para facilitar o consumo e a navegação entre os dados fornecidos em uma base de conhecimento.

3.8.1 Sistema de Percepção Humana

Entender como o sistema de percepção humana funciona nos ajuda a identificar quais visualizações de dados ou gráficos são mais efetivos na sumarização, exposição e compreensão dos dados (Bennett *et al.*, 2007; Graham, 2008; Kobourov *et al.*, 2015). Aspectos da visualização, como tipo de gráfico, cor, ou estética, também influenciam a carga de trabalho cognitivo de uma visualização e como ela é percebida (Borgo *et al.*, 2012; Hullman *et al.*, 2011; Moere *et al.*, 2012).

Enxergamos como resultado de um processo evolutivo que permitiu aos organismos detectar e interpretar a luz (Gross *et al.*, 2008). A visão humana tem a capacidade de detectar, perceber e processar o ambiente por meio da luz, utilizando os olhos e o sistema visual (Gross *et al.*, 2008). A capacidade de enxergar é produto de uma reação químico-elétrica que envolve a detecção de luz por fotorreceptores na retina, que convertem a luz em sinais elétricos enviados ao cérebro (Tovée, 2008). O cérebro interpreta esses sinais, permitindo a percepção de formas, cores, profundidade e movimento. A [Figura 28](#) ilustra parte deste processo. A visão é fundamental para a interação com o mundo, influenciando a navegação, a comunicação e a tomada de decisões.

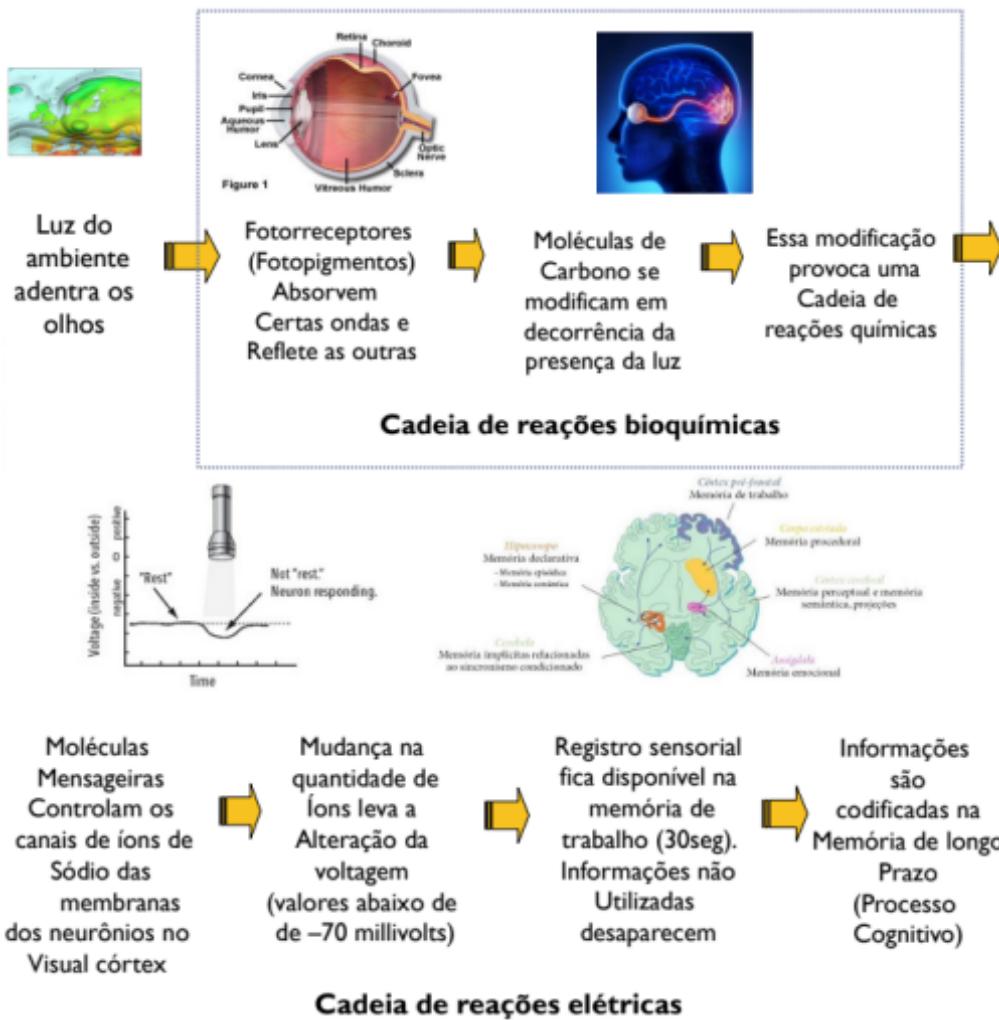


Figura 28: O processo biológico do sistema visual humano de acordo com Josko, 2023.

A forma como a informação é processada, escaneada pelos olhos e compreendida pelo ser humano é importante na elaboração de ferramentas de visualização de dados e exposição de informações (Pernice *et al.*, 2010). A Figura 29 mostra um tipo de padrão de leitura de telas.

3.8.2 Técnicas de visualização

“Excellence in statistical graphics consists of complex ideas communicated with clarity, precision, and efficiency.” Tufte & Graves-Morris (1983)

As visualizações de dados, diagramas de dados ou gráficos são organizados em taxonomias (Monmonier, 1985). Dentro da comunidade acadêmica de visualização de dados, existem muitas abordagens para a criação de taxonomias de visualização de dados (Rodrigues *et al.*, 2006). Tradicionalmente, muitas taxonomias de visualização foram baseadas em modelos de percepção gráfica, no layout visual e organizacional, bem como nas codificações gráficas dos dados apresentados (Shneiderman, 2003; von Engelhardt, 2002).

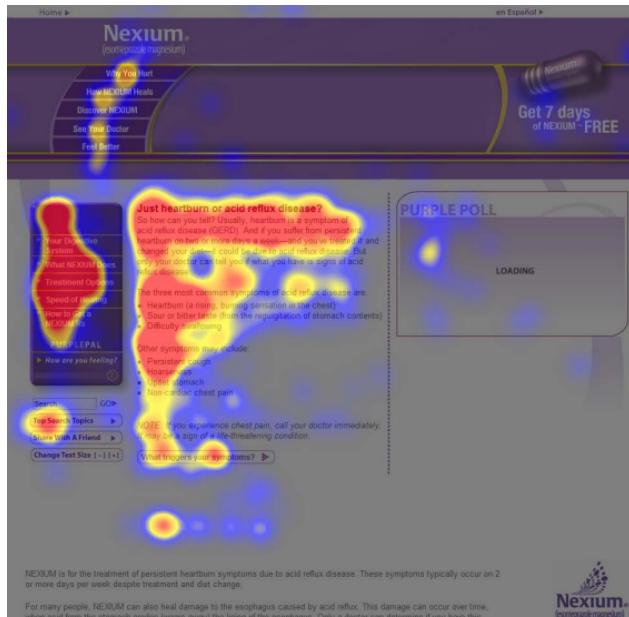


Figura 29: A imagem acima foi retirada do trabalho de Pernice *et al.* (2010) e mostra o mapa de calor formado pelas áreas mais vistas por usuários de um *website*, foi construído com dados de *eyetracking*. Observa-se o padrão F de leitura expresso pelos usuários no processamento da informação digital.

Embora existam muitas taxonomias para representar a vasta capacidade humana de representação visual de dados, neste trabalho o foco é em representações visuais para dados multidimensionais em mineração de dados.

Keim (1997); Keim & Kriegel (1996) dividiram as técnicas de visualização para a exploração de dados multidimensionais em seis classes: geométricas, baseadas em ícones, orientadas a *pixels*, hierárquicas, técnicas baseadas em grafos e híbridas. Neste trabalho, será adotado esta taxonomia, que é classificada em quatro grandes categorias de acordo com as abordagens gerais tomadas para gerar visualizações (De Oliveira & Levkowitz, 2003): projeção geométrica, técnicas orientadas a *pixels*, exibição hierárquica e iconografia.

As técnicas geométricas de visualização visam encontrar projeções e transformações informativas de conjuntos de dados multidimensionais, mapeando os atributos para um plano cartesiano de espaço arbitrário com coordenadas paralelas (Keim & Kriegel, 1996). Um exemplo de técnica geométrica é o gráfico de dispersão de dados em dois eixos cartesianos.

As técnicas orientadas a *pixels* funcionam representando um valor de atributo como um pixel na representação de uma visualização. Contudo, cada pixel pode ter propriedades, como cor, tamanho e formato. Um mapa de calor e um gráfico de preenchimento de espaço são bons exemplos deste tipo de técnica.

As técnicas de disposição hierárquica ou grafos tratam os dados de forma hierárquica, e sua representação contempla essas relações de hierarquia, que podem se dar por meio de relações entre atributos, relações entre valores de atributos ou entre valores de atributos e atributos. Um exemplo de técnica que utiliza essa abordagem é a representação visual de dados em árvores ou dendrogramas.

A iconografia é a técnica que se baseia em ícones, na qual cada valor de um atributo ou registro é mapeado com um ícone, um símbolo que representa o determinado registro e atribui alguma característica à representação visual. Para este tipo de técnica, bons exemplos são as faces de Chernoff ou mapas que possuem a informação iconográfica da demografia.

As visualizações de dados costumam utilizar mais de uma técnica de visualização, combinando-as em formas, formatos e características que atendem as necessidades das representações visuais. Da combinação de técnicas de visualização de dados dá-se o nome de visualizações híbridas (Chan, 2006; Tufte & Graves-Morris, 1983, c.7). Nas visualizações híbridas, é possível a utilização de cores, tamanho, comprimento, ícones, coordenadas, curvatura, densidade, textura, áreas, superfícies, sobreposição, *pixels*, linhas, formas e formatos, sombreamento, transparência, filtros, aproximação, afastamento, anotações, entre outras propriedades necessárias na representação multidimensional de dados (De Oliveira & Levkowitz, 2003; Tufte & Graves-Morris, 1983).

3.8.3 Visualizações de Dados

”Graphics reveal data. Indeed graphics can be more precise and revealing than conventional statistical computations.” Tufte & Graves-Morris (1983)

Em todas as representações visuais descritas, existem técnicas subjacentes que podem contribuir para facilitar a compreensão e a leitura dos gráficos, como:

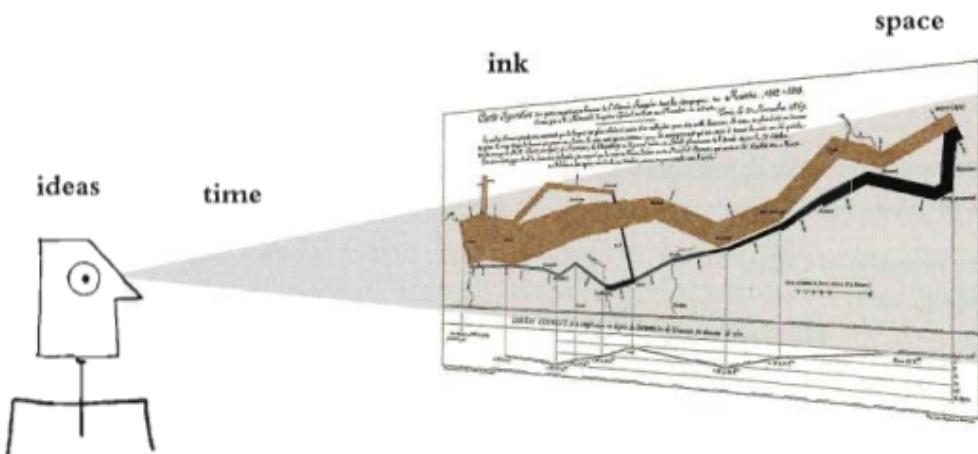
Anotações podem ser usadas para destacar, contextualizar e guiar a interpretação de informações específicas. As anotações são frequentemente usadas para chamar a atenção para valores ou tendências específicas, como picos, quedas ou valores discrepantes. Além disso, podem trazer explicações sobre a metodologia, eventos externos que influenciaram os dados ou interpretações que não são imediatamente evidentes apenas pela análise visual. As anotações podem ser textuais, incluindo rótulos, descrições ou explicações diretamente no

Principles of Graphical Excellence

Graphical excellence is the well-designed presentation of interesting data—a matter of *substance*, of *statistics*, and of *design*.

Graphical excellence consists of complex ideas communicated with clarity, precision, and efficiency.

Graphical excellence is that which gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space.



Graphical excellence is nearly always multivariate.

And graphical excellence requires telling the truth about the data.

Figura 30: Imagem adaptada do livro de Tufte & Graves-Morris, p.51, na imagem o autor descreve os princípios dos gráficos de excelência para visualizações gráficas baseadas em dados informacionais.

gráfico. Podem haver também anotações com ícones ou figuras. Elementos gráficos, como setas, círculos ou destaque coloridos, podem ser usados para direcionar a atenção a áreas específicas do gráfico. As anotações podem ser interativas, aparecendo somente quando os leitores de um gráfico interagem com os elementos gráficos, como *tooltips*.

Cores podem ser usadas para incorporar informações categóricas ou representar dimensões numéricas, indicando intensidade ou magnitude, como em gradientes de cores em uma escala gradual. As cores podem facilitar muito a leitura de gráficos. Normalmente, são selecionadas em uma paleta de cores que consiste em um conjunto predefinido de cores, selecionadas para serem usadas em um contexto específico, trazendo consistência e harmonia para as representações visuais.

Ícones podem ser utilizados para substituir valores dispostos em uma representação visual e podem indicar símbolos, áreas, formas ou objetos. O tipo, tamanho e cor dos ícones podem ajudar a transmitir informações de categorias, intensidade, magnitude, volume, entre tantas outras informações.

Interatividade torna as representações visuais mais dinâmicas, pois permite a atualização e o movimento dos componentes gráficos, como a utilização de animações, efeitos a medida que o leitor interage com o gráfico. Zoom é um tipo de interatividade e permite que os dados em uma representação visual possam ser ampliados ou reduzidos em áreas específicas de interesse. Outro exemplo são os filtros, que se tratam de condições aplicadas aos dados da representação visual, selecionando os dados que serão utilizados para renderização a medida que o usuário interage com o gráfico.

Gráficos de linha são representações visuais que conectam pontos de dados com uma linha, sendo ideais para mostrar a evolução dos valores ao longo do tempo. Este tipo de gráfico utiliza o eixo horizontal (X) para representar uma dimensão numérica, como o tempo, enquanto o eixo vertical (Y) mostra os valores de interesse. A conexão dos pontos facilita a identificação de mudanças e permite a comparação de várias séries de dados em um único gráfico.

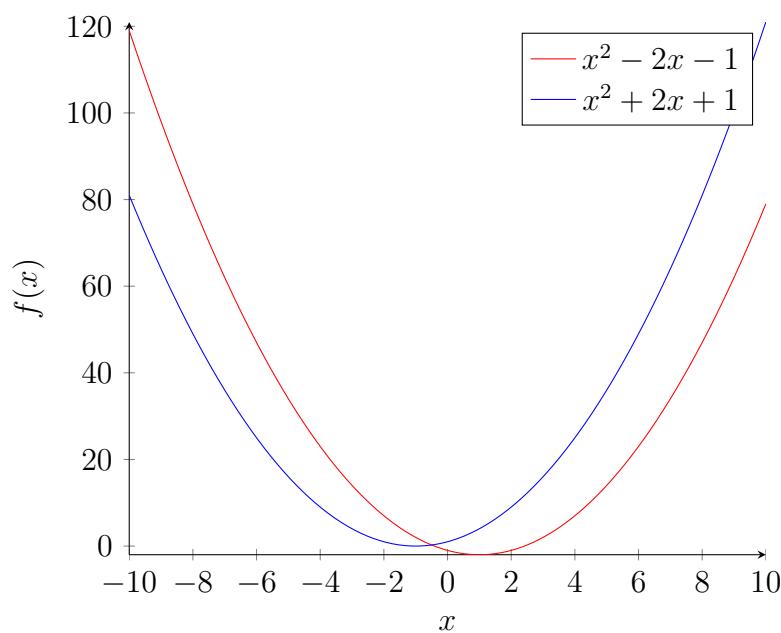


Figura 31: Um exemplo de gráfico de linha, perceba que há anotações, cores e mais de uma linha no gráfico, combinando diferentes técnicas de visualização.

Gráficos de barra são representações visuais que utilizam barras retangulares para comparar diferentes categorias ou grupos de dados. Normalmente, o eixo horizontal (abscissa X) exibe as categorias, enquanto o eixo vertical (ordenada Y) mostra os valores correspondentes. As barras podem ser verticais ou horizontais, dependendo da apresentação dos dados. Esses gráficos são eficazes para destacar comparações, e visualizar rapidamente qual categoria possui um valor maior ou menor. O comprimento ou altura das barras é proporcional ao valor que representam, e diferentes cores podem ser usadas para distinguir entre categorias ou grupos de interesse.

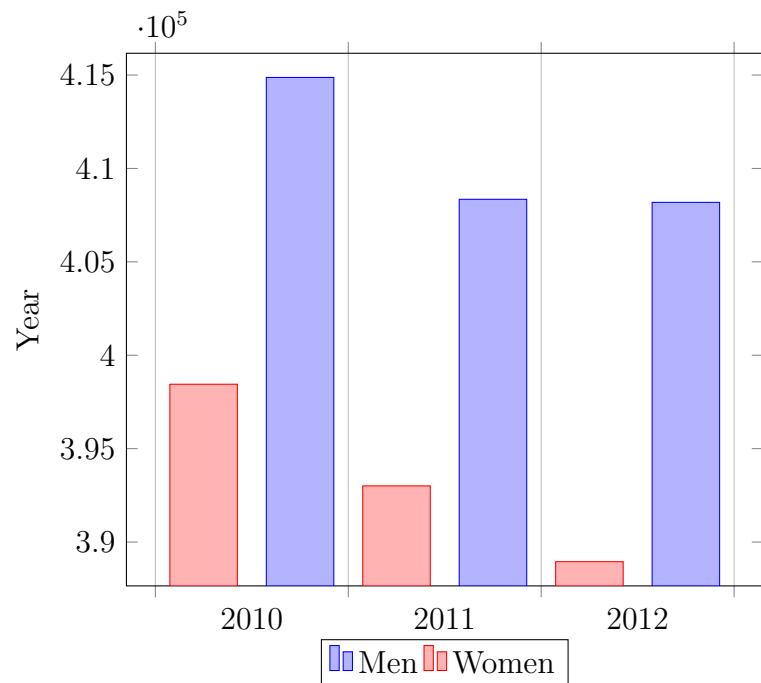


Figura 32: Um exemplo de gráfico de barra comparando dois grupos (cores) ao longo do tempo (eixo X) numa dimensão numérica (eixo Y).

Gráficos de pontos são representações visuais que utilizam pontos em um sistema de coordenadas bidimensional para mostrar a relação entre variáveis. O eixo horizontal (abscissa X) representa uma variável, enquanto o eixo vertical (ordenada Y) representa outra. Esses gráficos são simples e eficazes para visualizar grandes volumes de dados, facilitando a identificação de dispersões e valores discrepantes. Além disso, podem ser utilizados para comparar diferentes grupos, especialmente quando os pontos são diferenciados por cores, formas ou ícones.

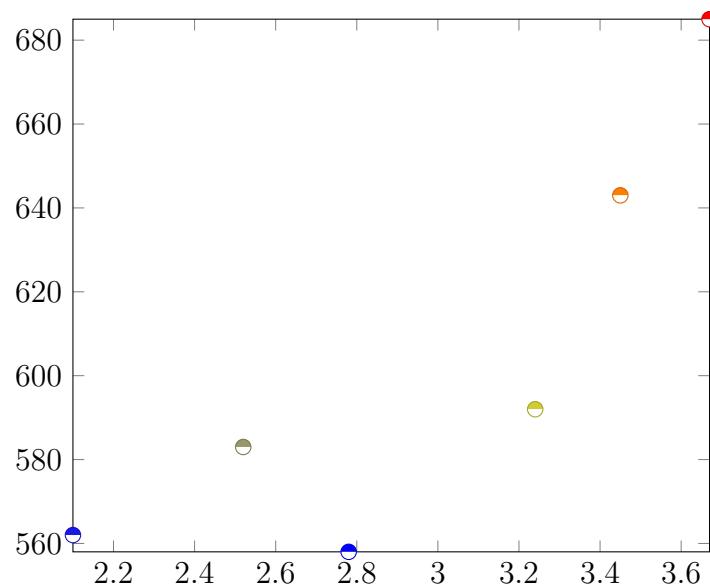


Figura 33: Um exemplo de gráfico de pontos, onde observamos a combinação de cores, ícones e duas dimensões numéricas.

Gráficos de área são uma forma de visualização de dados que combina elementos de gráficos de linhas e gráficos de barras. Este tipo de representação visual representa dados por meio de uma linha que conecta pontos, com a área abaixo preenchida. Esses gráficos são eficazes para comparar várias séries de dados, pois permitem que áreas diferentes sejam empilhadas ou sobrepostas. Eles ajudam a visualizar tendências, mostrando como os valores aumentam ou diminuem quando comparados ao eixo horizontal (abscissa X), e oferecem contexto visual sobre a magnitude dos dados em relação ao total.

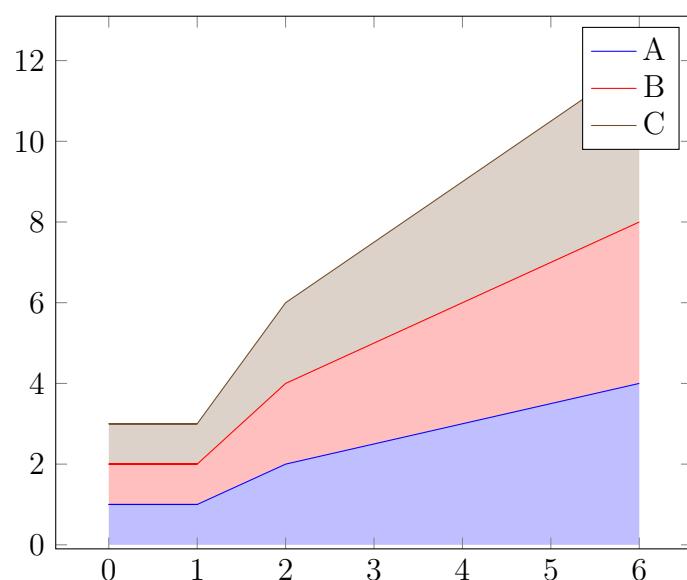


Figura 34: Um exemplo de gráfico de áreas, perceba a combinação de diferentes tipos de visualizações e técnicas.

Gráficos de setores são representações visuais que mostram a proporção de diferentes partes em relação a um todo. A figura do todo é dividida em fatias ou setores, onde cada fatia representa uma categoria ou segmento dos dados, permitindo visualizar a distribuição percentual entre diferentes segmentos do diagrama. Estes tipos de gráficos são intuitivos e fáceis de interpretar, sendo eficazes para mostrar como as partes se relacionam com o todo. No entanto, eles podem se tornar confusos com muitas categorias ou pequenas diferenças entre proporções.

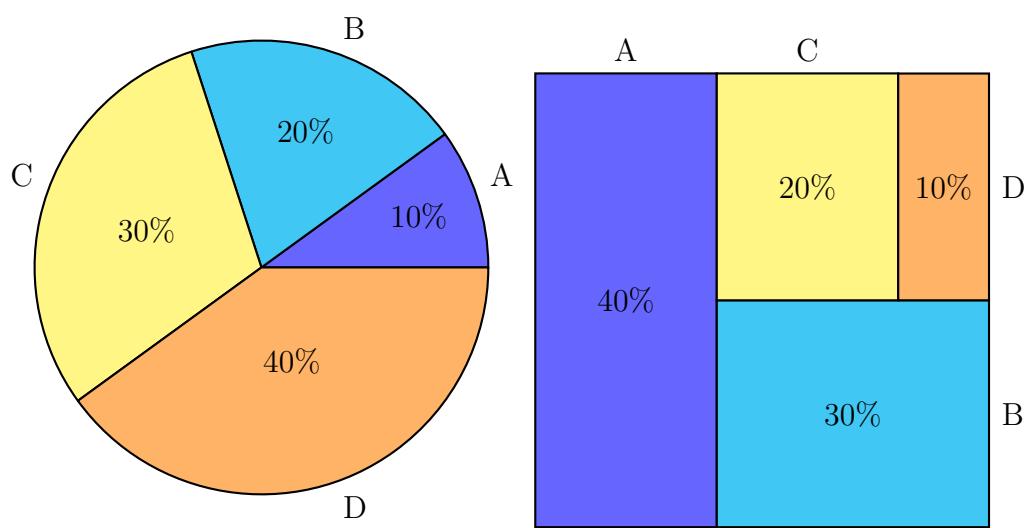


Figura 35: Um exemplo de gráfico de setores

Gráficos de Grid ou Matriz são representações visuais que organizam dados em uma estrutura de linhas e colunas, permitindo a visualização de relações entre múltiplas variáveis. Cada célula da matriz pode representar um valor específico, facilitando a análise de padrões e interações.

	1	2	3	4	5	6	7	8	9
a	74	25	39	20	3	3	3	3	3
b	25	53	31	17	7	7	2	3	2
c	39	31	37	24	3	3	3	3	3
d	20	17	24	37	2	2	6	5	5
e	3	7	3	2	12	1	0	0	0
f	3	7	3	2	1	36	0	0	0
g	3	2	3	6	0	0	45	1	1
h	3	3	3	5	0	0	1	23	1
i	3	2	3	5	0	0	1	1	78

Figura 36: Um exemplo de gráfico matriz que representa um diagrama de mapa de calor, pois combina uma escala de cores, um gradiente que varia conforme a variação de uma variável numérica.

Diagramas de grafos são representações visuais que mostram as relações entre entidades, chamadas de vértices, conectados por arestas. Esses grafos são usados para modelar e analisar redes complexas, como redes sociais e sistemas de transporte. A estrutura não linear dos grafos permite representar relações complexas de forma intuitiva. Eles são úteis para analisar propriedades da rede, como centralidade e caminhos mais curtos. Esses grafos ajudam a identificar padrões em sistemas interconectados.

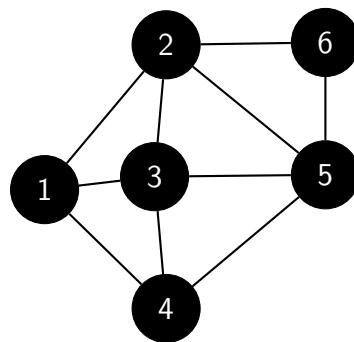


Figura 37: Um exemplo de grafo, traz uma relação hierárquica entre os pontos. Esse tipo de diagrama pode ser combinado com cores, tamanho e ícones para gerar visualizações complexas e com grande quantidade de informação.

Diagramas de nuvens de texto (ou *Word Cloud*) são representações visuais que utilizam caracteres ou palavras para transmitir informações de forma estruturada. Este tipo de gráfico combina texto com outras técnicas, como tamanho e cor, facilitando a compreensão ao usar palavras em vez de apenas números ou símbolos.

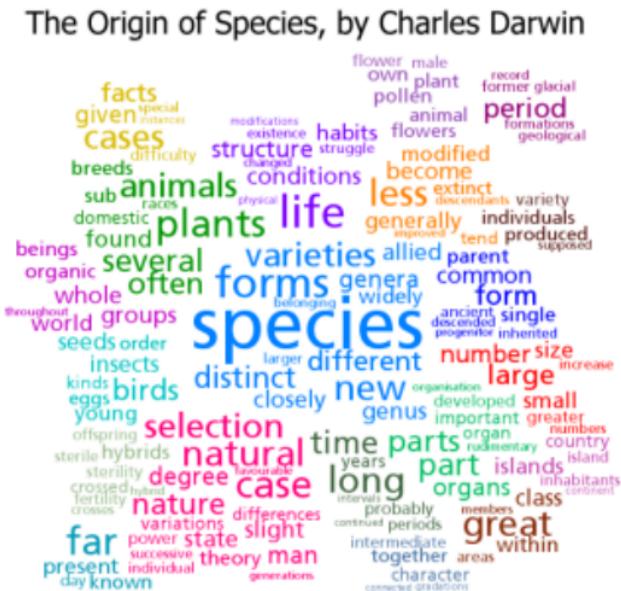


Figura 38: Um exemplo de gráfico de nuvem de palavras (*Word Cloud*), adaptado de Hearst *et al.* (2019). Perceba que palavras maiores possuem valores maiores, combina-se também cores ou ícones na visualização de dados.

Boa parte das técnicas e visualizações de dados descritas foi utilizada na ferramenta Summarticles. Esta teoria de visualização de dados fornece um conjunto de ferramentas gráficas que apoia a descoberta de conhecimento pelo usuário e a summarização gráfica das informações após o processamento dos dados.

4 Conjunto de dados utilizado

O conjunto de dados utilizado para o desenvolvimento do projeto foi constituído com base em artigos científicos públicos e de acesso livre, todos em formato de arquivos digitais PDF, obtidos por extração na *Web* por meio de um programa Python. Este programa Python captura os PDFs dos artigos científicos e os disponibiliza para realizar a extração do texto com a ferramenta Summarticles.

Foram extraídos cerca de 1000 artigos científicos por meio de um programa Python. Estes artigos foram extraídos de repositórios de artigos científicos ResearchGate, Science Direct e *Web of Science*. Os artigos extraídos não foram utilizados na sua totalidade, somente uma amostra de 100 artigos foi utilizada para o desenvolvimento da ferramenta.

Por se tratar de uma ferramenta que processa dados textuais de artigos científicos, a aplicação Summarticles não possui uma base de conhecimento específica para ser utilizada. Isto significa que qualquer artigo científico pode servir como entrada para a ferramenta. Isso facilita sua utilização e teste, dado que qualquer conjunto de artigos científicos pode ser processado pela ferramenta.

Nos testes com a ferramenta Summarticles, o número máximo de artigos científicos utilizados por processamento foi de 30 artigos científicos, no formato digital PDF, com uma média de 5 páginas por artigo. Artigos mais longos demoram mais para serem processados, pois a ferramenta consome o processamento da máquina local na qual está instalada. Portanto, utilizações da ferramenta com mais de 30 artigos científicos e artigos científicos longos podem aumentar significativamente o tempo de processamento e a geração das análises.

4.1 Dados Extraídos de Artigos

Durante a execução da ferramenta Summarticles todos os artigos passam pelo serviço da ferramenta GROBID que processa e extrai todos os dados não-estruturados dos artigos científicos. Texto, equações, diagramas e imagens são extraídos e convertidos em arquivos XML. Após o processamento dos artigos de entrada, Summarticles prepara e transforma os dados em um formato tabelar, possibilitando a geração de estatísticas, visualizações e a

entrada em algoritmos de aprendizado de máquina.

A extração de dados dos artigos científicos nem sempre é possível em sua totalidade. Por conta da heterogeneidade de formatos de estrutura dos artigos científicos, algumas informações não são extraídas, como o ano de publicação, a instituição de afiliação dos autores, país da afiliação, entre outras informações que podem não estar disponíveis, o que torna o processamento e as análises instáveis.

4.1.1 Metadados

Os dados obtidos de cada artigo científico são armazenados em estruturas semi-estruturadas e tabelas estruturadas. As estruturas de dados obtidas são:

1. Informações do cabeçalho:

- Título, data, instituição, publicador, nota, DOI, link da publicação, códigos como ISSN, EISSN e PMID.

2. Informações dos autores:

- Nomes, e-mail, ORCID, instituição, departamento, laboratório, código postal, província e país da publicação;

3. Informações do corpo do artigo:

- Resumo, agradecimentos, língua do artigo, anexos e conteúdo textual do corpo do artigo.

4. Informações de referências e citações:

- Títulos, nome, instituição, laboratório, país, departamento, páginas, DOI, códigos como PMID, PMCID, ARXIV-ID, ARK, url, entre outras informações de citações.

Todas essas informações podem ser extraídas de cada artigo científico, embora, na prática, muitos desses campos não sejam extraídos devido à dificuldade na identificação de seus valores, dada a variedade de formatos dos artigos científicos.

5 Ferramenta Computacional Summarticles

A proposta da ferramenta Summarticles, disposta neste projeto é facilitar o acesso a análises bibliométricas de forma rápida, gratuita, *open source*, em uma abordagem *plug and play*, sem ser necessário ter um vasto conhecimento técnico ou realizar tratamentos manuais nos dados dos artigos científicos. Em suma, o processo executado pela ferramenta consiste na captura dos artigos científicos no formato .PDF e na geração de painéis com visualizações de fácil compreensão. O diagrama da [Figura 39](#) exemplifica o processo de execução da ferramenta Summarticles, desde a entrada, composta por um conjunto de artigos científicos, até a saída, que corresponde a um conjunto de visualizações analíticas.

A ferramenta Summarticles funciona em um ambiente local. Portanto, depende da capacidade computacional do dispositivo no qual a ferramenta está sendo executada. Não há aplicação de paralelismo em sua totalidade de processamento, mas alguns componentes são utilizados como serviços (Docker, Ollama, GROBID), habilitando algum grau de processamento paralelo. Contudo, boa parte dos componentes (geração de gráficos, algoritmos de inteligência artificial) da ferramenta funciona de maneira sequencial, com execução não paralela.

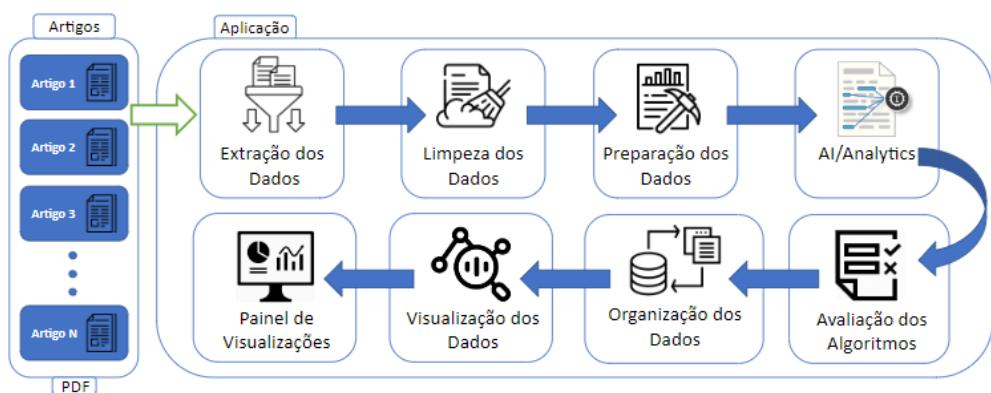


Figura 39: Processo simplificado de execução da ferramenta *Summarticles*.

5.1 Ambiente de Desenvolvimento

A ferramenta Summarticles foi desenvolvida na linguagem de programação Python, utilizando diversas bibliotecas para manipulação de dados, processamento de texto, visualização de dados e criação de modelos de aprendizado de máquina. Além disso, foi utilizada a pla-

taforma GROBID para a extração de dados de artigos científicos, a plataforma Docker para virtualização de serviços e a plataforma Ollama para a execução de grandes modelos de linguagem.

5.1.1 Dispositivo utilizado

As fases de desenvolvimento, testes e uso da ferramenta Summarticles foram realizados nos sistemas operacionais Linux 25.10 e Windows 11. Foi utilizado um dispositivo de 32 *gigabytes* de memória RAM, com um processador *Ryzen 7 5700x3d* de 8 núcleos, 16 *threads* e 3.0 *gigahertz*; a aplicação necessita de 10 *megabytes* de armazenamento.

Recomenda-se a utilização da ferramenta em dispositivos com mais de 8 *gigabytes* de memória RAM e com processadores superiores a 4 núcleos, 2.0 *gigahertz*, nos sistemas operacionais Linux 25.10 e Windows 11, ou versões superiores. A aplicação não foi testada em outras configurações de dispositivos não mencionadas neste projeto. Por esse motivo, a utilização e os testes em outros ambientes ficam a cargo do usuário da ferramenta Summarticles.

5.1.2 Docker

A plataforma Docker ([Merkel, 2014](#)) é uma plataforma de código aberto que permite virtualizar aplicações de software e suas dependências em ambientes isolados, leves e portáveis, chamados de contêiners. Os contêiners possuem todo o código e as dependências necessárias para rodar as aplicações de forma isolada, permitindo que haja consistência na execução das aplicações de software em diferentes ambientes. Neste trabalho, a ferramenta Docker suporta o serviço do GROBID e Ollama.

5.1.3 GROBID

Utilizamos a ferramenta GROBID ([Lopez, 2008–2025](#)) para a extração de informações dos artigos científicos. O nome GROBID é uma abreviação de *GeneRation Of Bibliographic Data*. O GROBID trata-se de uma biblioteca em Java, de aprendizado de máquina, com software aberto, desenvolvida para extrair, analisar e reestruturar documentos brutos, como PDFs, em documentos estruturados codificados, com foco particular em publicações técnicas

e científicas.

5.1.4 Ollama

Ollama (Ollama, 2023) é uma plataforma de código aberto que permite executar, gerenciar e interagir com grandes modelos de linguagem de forma fácil, utilizando recursos locais de um computador, ou seja, sem necessitar de comunicação com APIs pagas dos serviços de LLMs em nuvem. Utilizamos Ollama para executar e gerenciar os modelos Llama 3 e Deepseek-R1.

5.1.5 Python

Python (Van Rossum & Drake, 1989) é uma linguagem de programação de alto nível, interpretada, multiparadigma e de código aberto. Ela foi criada por Guido van Rossum e lançada pela primeira vez em 1991. Esta foi a linguagem de programação utilizada para a elaboração do *software* e a execução dos algoritmos. A versão do *Python* utilizada foi a 3.11.

As bibliotecas usadas neste projeto de pesquisa são:

Scikit-learn (Pedregosa *et al.*, 2011) é uma biblioteca de código aberto utilizada para tarefas de aprendizado de máquina em *Python*, que fornece uma ampla variedade de algoritmos e ferramentas para construir, treinar e avaliar modelos de aprendizado de máquina em tarefas como classificação, regressão, agrupamento e muito mais.

Pandas (pandas development team, 2020) é uma biblioteca de código aberto utilizada para a manipulação e análise de dados. Ela oferece estruturas de dados flexíveis, como *DataFrames* e *Series*, que permitem a organização, limpeza e análise de dados tabulares.

NumPy (Harris *et al.*, 2020) é uma biblioteca *Python* de código aberto que oferece recursos para trabalhar com *arrays* multidimensionais (vetores, matrizes e afins) e realizar cálculos matemáticos de alta performance, tornando a manipulação de dados numéricos mais eficiente.

Foram utilizadas as bibliotecas de código aberto para visualização de dados: *Plotly*, *Matplotlib*, *HoloViews*, *Seaborn* e *GraphViz*. *Plotly*, *HoloViews* e *GraphViz* para visualizações dinâmicas; *Matplotlib* e *Seaborn* para visualizações estáticas.

O *LangChain* é uma plataforma de código aberto projetado para facilitar a criação de

aplicações baseadas em grandes modelos de linguagem.

Além das bibliotecas mencionadas acima, existem outras utilizadas no desenvolvimento do projeto, todas as bibliotecas usadas serão documentadas e disponibilizadas no repositório de código da ferramenta.

5.1.6 Streamlit

Streamlit ([Streamlit Inc., 2026](#)) é uma biblioteca de código aberto para a linguagem de programação *Python*, sua finalidade é permitir que desenvolvedores transformem facilmente programas *Python* em aplicativos *Web* funcionais, possibilitando a criação de interfaces de usuário interativas para análises de dados, visualizações, protótipos e outras aplicações.

5.2 Diagramas da Aplicação Summarticles

Os diagramas nos ajudam a entender como a ferramenta funciona, evidenciando seus componentes, integrações e a ordem de execução. Cada componente possui características específicas, e possui uma função para realizar o objetivo da ferramenta.

5.2.1 Diagrama de Componentes

O diagrama [Figura 40](#) traz os componentes da ferramenta Summarticles e como ocorrem suas integrações:

- `Summarticles.py`: o componente principal da ferramenta é a sua interface *Web*. Ele é um aplicativo sustentado pela aplicação Streamlit que executa em um navegador. Essa interface de aplicação *Web* é a interface pela qual o usuário interage com toda a aplicação e usa as funcionalidades da ferramenta Summarticles, desde a entrada dos dados até a saída das visualizações de gráficos. O módulo `summarticles.py` contém toda a lógica da aplicação e integra-se a todos os outros módulos para orquestrar a aplicação.
- `Grobid-client.py`: um outro componente é o contêiner da plataforma Docker que executa o serviço do GROBID. Summarticles interage com esse serviço por meio do módulo `grobid-client.py`. Este módulo envia tarefas e requisições para o serviço do GROBID.

- `Summacomponents.py`: renderiza todos os componentes da página Web. Ela contém também a lógica por trás dos menus, botões e componentes HTML.
- `Summaviz.py`: é responsável por gerar todas as visualizações de dados e renderizá-las na aplicação Web.
- `Text.py`: realiza tarefas de processamento de texto, utilizando técnicas de mineração de texto, preparação e execução dos códigos de alguns algoritmos de *machine learning* aplicados aos textos de artigos científicos.
- `Summachat.py`: realiza todo o processo de execução do chat conversacional na plataforma. Ela realiza o processo de comunicação com as LLMs na plataforma Ollama ou APIs on-line, renderiza o componente de chat na aplicação Web e faz o processo de RAG.
- `Summaetl.py`: possui funções para tratamento de dados tabulares e grafos.
- `Summautils.py`: contém funções utilitárias que são utilizadas em diversos módulos.

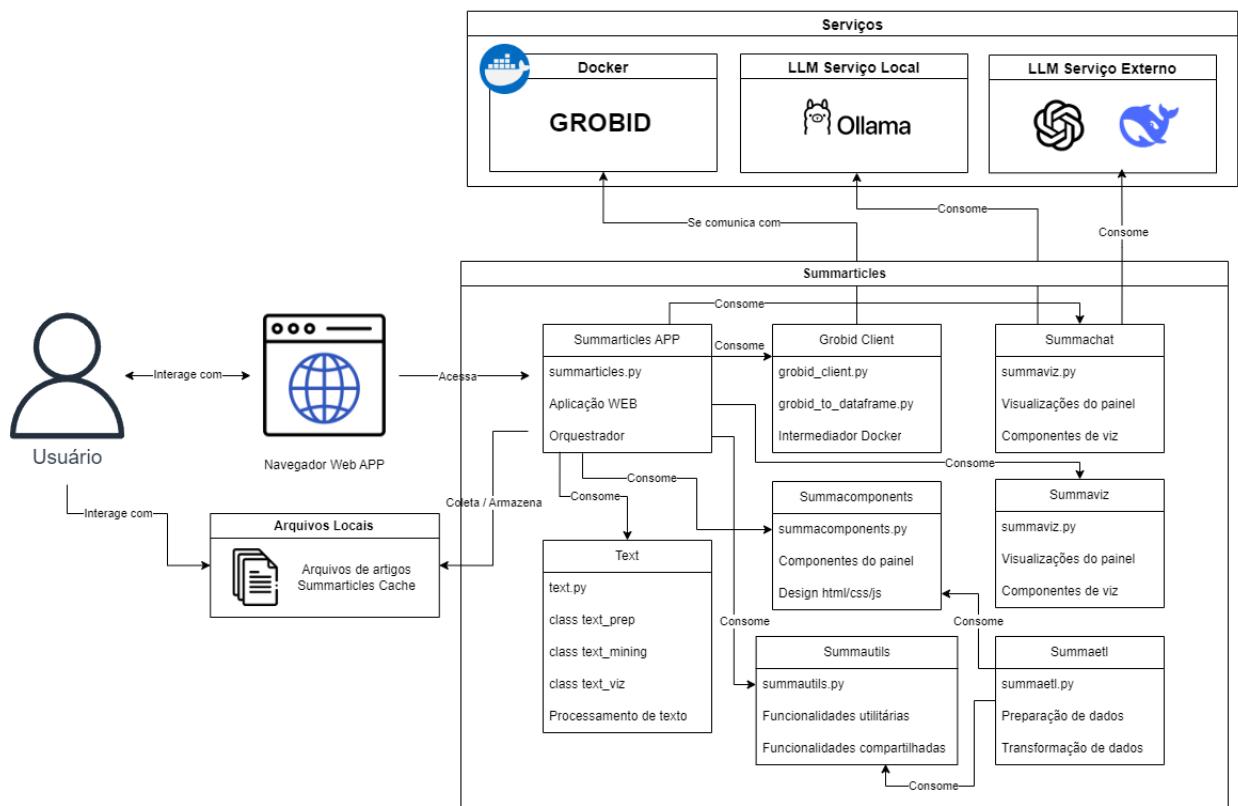


Figura 40: Arquitetura de componentes da ferramenta Summarticles em alto nível.

5.3 Utilizando a ferramenta Summarticles

5.3.1 Instalação da ferramenta Summarticles

A instalação da ferramenta Summarticles passa por quatro passos:

1. Plataforma Python 3.11:

- Instale o plataforma Python, disponível em [?;](#)
- Abra um terminal de comando;
- Execute o comando: "python –version";
- Verifique se o resultado do comando anterior corresponde à versão necessária para executar a plataforma (Python 3.11).

2. Plataforma Docker:

- Instale a plataforma Docker, disponível em [Merkel \(2014\);](#)
- Execute o plataforma Docker;
- Abra um terminal de comando;
- Execute o comando: "docker run -t –rm –init -p 8080:8070 -p 8081:8071 lfoppi-ano/grobid:0.7.0".

3. Plataforma Ollama:

- Instale a plataforma Ollama, disponível em [Ollama \(2023\);](#)
- Execute o software Ollama;
- Abra um terminal de comando;
- Execute o comando: "ollama pull deepseek-r1:1.5b";
- Execute o comando: "ollama pull llama3.2:1b".

4. Repositório da ferramenta Summarticles:

- Baixe a ferramenta Summarticles, disponível em [Batista \(2025\);](#)
- Descompacte o arquivo "summarticles.zip" no local desejado;

- Abra um terminal de comando;
- Navegue pelo terminal de comando até o repositório local da ferramenta;
- Execute o comando: "python -m venv summa";
- No sistema operacional Windows execute o comando: "./summa/Scripts/activate". No sistema operacional Linux execute o comando: "source summa/bin/activate";
- Execute o comando: "pip install --no-cache-dir -r summarticles/requirements.txt"

Após a execução desses passos, a ferramenta está instalada e disponível para ser executada. Para ambientes linux, deve-se executar o comando "sudo apt install python3-tk".

5.3.2 Executando a Ferramenta Summarticles

Para executar a ferramenta Summarticles, após a instalação, basta seguir os seguintes passos:

1. Abra um terminal;
2. Navegue pelo terminal de comandos até o repositório local da ferramenta;
3. No repositório que contém o arquivo "summarticles.py";
4. No sistema operacional Windows execute o comando: "./summa/Scripts/activate". No Linux execute o comando: "source summa/bin/activate";
5. Execute o comando: "streamlit run summarticles.py";
6. Acesse no navegador o endereço <http://localhost:8501/>;

Após executar os passos acima, basta utilizar a ferramenta, fornecendo um diretório com artigos científicos no formato PDF como entrada.

5.3.3 Entrada da Ferramenta Summarticles

A ferramenta Summarticles recebe como entrada um caminho de diretório para uma pasta contendo arquivos em formato PDF, como pode ser visto na imagem [Figura 41](#).

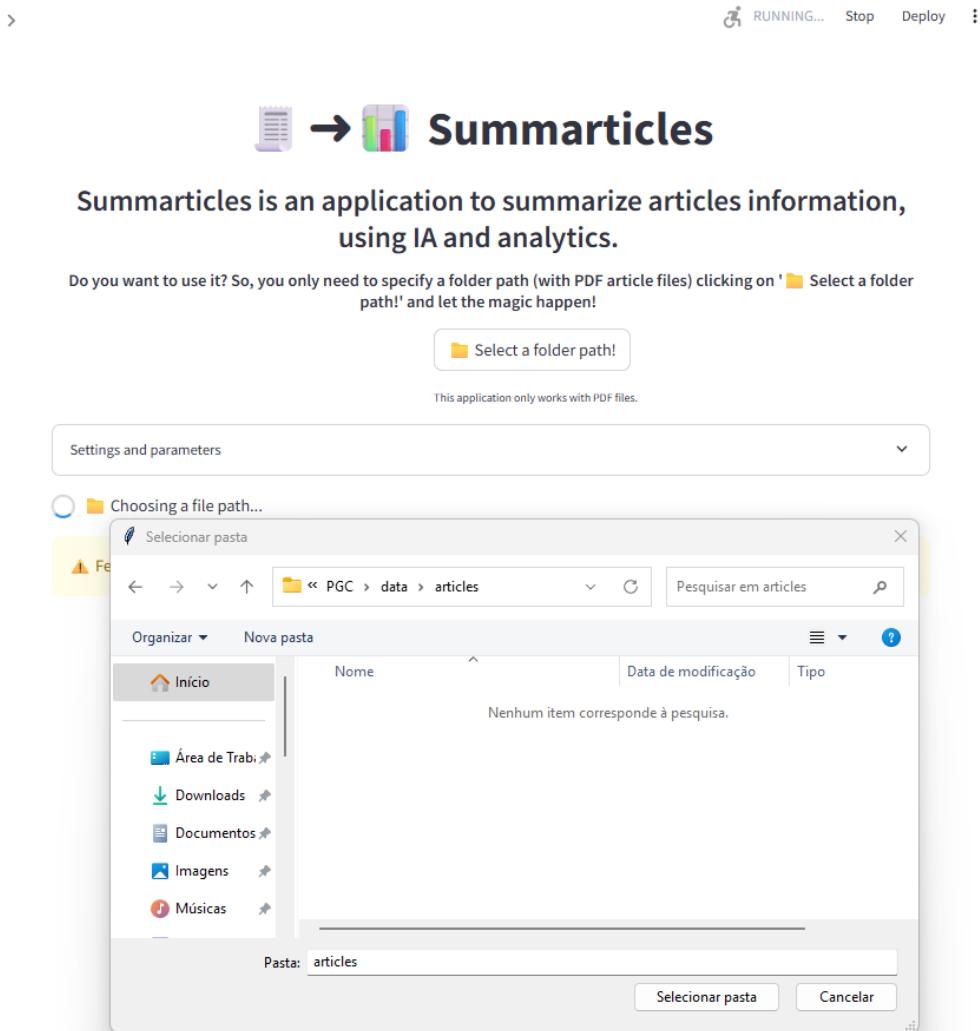


Figura 41: Acionando o botão ”Select a folder path!” uma janela de seleção é aberta possibilitando a busca do diretório de entrada.

Abaixo segue uma lista com um conjunto de artigos científicos que poderiam ser usados como entrada na ferramenta Summarticles:

1. Deep Residual Learning for Image Recognition:

- Disponível em [He et al. \(2015\)](#);

2. Using thematic analysis in psychology:

- Disponível em [Braun & Clarke \(2006\)](#);

3. Random forests a Machine Learning Algorithm:

- Disponível em [Breiman \(2001\)](#);

4. Attention Is All You Need:

- Disponível em [Vaswani et al. \(2023\)](#);

5. ImageNet classification with deep convolutional neural networks:

- Disponível em [Krizhevsky et al. \(2012\)](#);

Em suma, para o funcionamento da ferramenta, basta ter como entrada um repositório com um conjunto de artigos científicos dispostos em arquivos no formato PDF.

5.3.4 Saída da ferramenta Summarticles

A saída da ferramenta Summarticles corresponde a um conjunto de estatísticas e visualizações interativas em uma página que pode ser visualizada em um navegador *Web*. O endereço de acesso da saída da ferramenta é "<http://localhost:8501/>". A página da ferramenta Summarticles pode ser salva utilizando as funções do navegador.

Summarticles também armazena um conjunto de arquivos temporários de suas execuções como cache. Estes arquivos temporários são utilizados para recuperar relatórios e acelerar o processamento dos artigos que já foram processados anteriormente. Os arquivos de cache ficam disponíveis no mesmo repositório de entrada, com o nome "summarticles-cache".

6 Comparação de Ferramentas

Na [Tabela 12](#) é mostrada uma comparação entre as ferramentas citadas, considerando os critérios discutidos nos parágrafos abaixo. Cada coluna da tabela identifica um critério e, cada linha, uma ferramenta discutida: a coluna ”Grátis” indica se a ferramenta é gratuita, isto é, se não há nenhum gasto financeiro para sua utilização; a coluna ”Aberta” indica se a ferramenta possui código aberto para a comunidade; a coluna ”Programar” indica se há necessidade prévia de ter conhecimento em programação para a utilização da ferramenta; a coluna ”Pré-Processa” indica se é necessário realizar algum tipo de tratamento nos dados para a sua utilização; a coluna ”Requisito” indica se há algum pré-requisito ou dependência para o uso da ferramenta; por fim, a coluna ”Configurar” indica se é necessária alguma configuração de ambiente ou sistema operacional para o uso da ferramenta.

	Grátis	Aberta	Programar	Pré-Processa	Requisito	Configurar
Summarticles	Sim	Sim	Não	Não	Sim	Sim
VOSviewer	Sim	Sim	Não	Sim	Sim	Não
Gephi	Sim	Sim	Não	Sim	Sim	Não
SciMAT	Sim	Sim	Não	Sim	Sim	Não
CiteSpace	Não	Sim	Não	Sim	Sim	Não
Bibliometrix	Sim	Sim	Sim	Sim	Sim	Sim
Leximancer	Não	Não	Não	Não	Não	Não
Scholarcy	Não	Não	Não	Não	Não	Não
Sciwing	Sim	Sim	Sim	Sim	Sim	Sim

Tabela 12: Comparação entre ferramentas

A ferramenta *VOSviewer* é uma ferramenta com ênfase em visualizações de redes científicas, direcionada para estudos bibliométricos. Ela permite mapear relações entre autores, artigos, instituições e mapas de co-ocorrência. *Gephi* é muito semelhante à ferramenta *VOSviewer*, mas seu foco está mais direcionado para a visualização de redes complexas e operações em grafos, como pesquisas em redes biológicas, redes sociais e cálculo de métricas em grafos, como *clusters*, e medidas de centralidade. *SciMAT* é muito similar à *VOSviewer*, *CiteSpace* e *Gephi*, possui análise de redes complexas, porém traz visualizações da evolução temática de um tema ao longo do tempo, identificando tendências. *VOSviewer*, *Gephi*, *SciMAT* e *CiteSpace*, são muito semelhantes, oferecendo funcionalidades sobrepostas, a interface e o uso semelhante.

Bibliometrix é um pacote em R especializado em análise bibliométrica. Ela permite pro-

cessar dados de bases como Scopus, *Web of Science*, e possui inúmeras funções estatísticas e visualizações. Porém, ela necessita de conhecimento na linguagem de programação R, e seu uso passa pelo desenvolvimento de código.

Leximancer possui seu foco no mapeamento de relações entre documentos, utilizando análise de texto e inteligência artificial para criar mapas visuais de conexão entre documentos, conceitos e palavras-chave. Muito usado para pesquisas qualitativas e análise de conteúdo.

Scholarcy automatiza a análise e o resumo de artigos, extraíndo e processando dados textuais com algoritmos de inteligência artificial. Seu foco está na extração de dados com qualidade e na criação de um resumo de dados textuais, e não há uso de visualizações ou análise gráfica. Sciwing é uma ferramenta similar à Scholarcy. Porém, é gratuita e flexível, focada em processamento e análise de documentos científicos, com funcionalidades de extração de informações e sumarização automática de texto.

Summarticles oferece visualizações analíticas interativas, como agrupamento de documentos, similaridade entre documentos, visualizações em grafos e a possibilidade de interagir com documentos por meio de grandes modelos de linguagem. Contudo, oferece mais requisitos para uso e uma necessidade de configuração maior do que as outras ferramentas comparadas. Seu foco está direcionado a uma análise textual, visualizações analíticas e na utilização de modelos de inteligência artificial para facilitar a descoberta de conhecimento. Existe a necessidade de instalação e configuração de outros softwares para a utilização da ferramenta Summarticles, e embora a instalação de softwares pelo terminal com comandos seja mais fácil do que a necessidade de saber alguma linguagem de programação, essa necessidade pode dificultar o uso da ferramenta por alguns usuários que não possuam esse conhecimento.

7 Considerações finais

Este Projeto de Graduação em Ciência da Computação tem como objetivo o desenvolvimento de uma ferramenta de software gratuita, denominada Summarticles, destinada à extração de informações, à sumarização de conteúdo, à geração de visualizações analíticas e à descoberta de conhecimento em conjuntos de artigos científicos. Conforme descrito, a ferramenta foi concebida para enfrentar o desafio imposto pelo crescente volume de publicações científicas e a constante necessidade de atualização pelos pesquisadores acerca de publicações científicas em diferentes campos de pesquisa. A ferramenta Summarticles busca facilitar o trabalho de cientistas e pesquisadores na descoberta de conhecimento, apoiando a análise bibliográfica.

A ferramenta Summarticles cumpre sua proposta ao extrair informações textuais de arquivos digitais em formato PDF e sumarizar boa parte do conhecimento neles contido. Para isso, emprega uma combinação de algoritmos de mineração de texto, algoritmos de inteligência artificial e a elaboração de visualizações analíticas interativas, disponibilizadas em uma interface web gratuita. O desenvolvimento da ferramenta também permitiu exercitar as competências e habilidades adquiridas durante a formação acadêmica em Ciência da Computação, aplicando conhecimentos de inúmeras disciplinas da graduação em Ciência da Computação.

O desenvolvimento destacou etapas críticas no processamento de artigos. O pré-processamento de texto mostrou-se essencial para transformar os dados não estruturados em representações estruturadas ou semi-estruturadas, como arquivos XML ou representações vetoriais (*Bag of Words* ou *embeddings*), permitindo, assim, a aplicação de algoritmos de aprendizado de máquina e inteligência artificial generativa.

Verificou-se que a qualidade da extração de dados, realizada pela plataforma GROBID, impacta diretamente a qualidade das análises e visualizações geradas. A heterogeneidade encontrada nos formatos dos artigos científicos foi um desafio, fazendo com que, em alguns casos, metadados importantes não fossem extraídos, o que afetou a estabilidade das análises e a qualidade da descoberta de conhecimento pelos pesquisadores e cientistas que podem utilizar a ferramenta. Ademais, observou-se que, como se trata de uma ferramenta de uso local, sua utilização torna-se muito dependente da capacidade de processamento do dispositivo em

que está sendo executada. Neste sentido, pode ser moroso para alguns dispositivos processar grandes volumes de artigos científicos e utilizar todas as suas funcionalidades.

A ferramenta também incorpora funcionalidades de IA generativa, utilizando grandes modelos de linguagem executados localmente via Ollama e aplicando a técnica de geração aumentada via recuperação. Isso permite que o usuário interaja com os documentos de forma conversacional, utilizando a coleção de artigos como base de conhecimento externa, uma funcionalidade não vista quando comparamos com outras ferramentas.

Portanto, embora a ferramenta possua um grau de configuração que depende de conhecimento, quando comparamos às demais utilizadas neste projeto, não há necessidade de ter conhecimento em alguma linguagem de programação para sua utilização. Podemos também observar que a Summarticles traz algumas outras funcionalidades analíticas e conversacionais. Finalmente, a ferramenta pode ser customizada, uma vez que ela é de código aberto e gratuita. Essas vantagens tornam a ferramenta Summarticles competitiva e funcional.

7.1 Limitações

Apesar de atingir os objetivos específicos propostos, a ferramenta Summarticles apresenta limitações. A principal delas é a necessidade de configuração do ambiente para a sua execução. Conforme detalhado na seção 5.3.1, o usuário precisa instalar e configurar múltiplos *softwares*, como Python, Docker (para o serviço GROBID) e Ollama (para os serviços locais de LLMs). Embora a ferramenta não exija conhecimento em programação para ser utilizada, essa configuração inicial pode representar uma barreira para usuários sem conhecimento técnico.

Outra limitação refere-se ao desempenho do processamento. Por ser uma aplicação que executa localmente, o tempo de análise depende da capacidade de processamento do dispositivo do usuário (*hardware*). Testes indicaram que o processamento de artigos longos ou de um volume superior a 30 artigos pode aumentar consideravelmente o tempo na geração das análises. Ademais, como muitas tarefas da ferramenta são processadas sequencialmente, o processo torna-se moroso quando há um grande volume de dados a ser processado.

Embora muitas das visualizações incorporadas na ferramenta sejam fáceis de entender, ainda há espaço para torná-las mais simples e de fácil compreensão. Ainda há oportunidades

para melhorias visuais, pois a dependência de múltiplos pacotes de visualização trouxe um aumento na complexidade da renderização das visualizações analíticas.

Muitos algoritmos de inteligência artificial requerem um poder computacional mais elevado, necessitando da utilização de unidades de processamento gráfico (ou GPUs). Para que se possa incorporá-los à ferramenta Summarticles, adotou-se versões simplificadas desses algoritmos ou versões truncadas dos dados. Ao realizar essas simplificações permitiu a utilização desses algoritmos em detrimento da qualidade das análises geradas por eles.

7.2 Trabalhos futuros

Para mitigar a dependência do GROBID e as falhas de extração de dados, futuras pesquisas podem ser direcionadas a novas ferramentas de extração de dados não-estruturados, como Docling, Textract, Marker, Nougat e MinerU. Novas ferramentas para a extração dos dados podem ajudar a aumentar a qualidade da extração e mitigar o efeito de heterogeneidade nos formatos de artigos científicos.

Diminuir a necessidade de configuração da ferramenta Summarticles é um ponto de evolução que será estudado no futuro, além de avaliar a possibilidade de utilização de um serviço *Web* ao invés de uma arquitetura local é um possível caminho. Além disso, será avaliada a possibilidade de compilar toda a ferramenta em um único *software*, com um único instalador para as configurações da ferramenta.

Por fim, dado o avanço da inteligência artificial generativa e a disponibilização de modelos de linguagem pré-treinados, as pesquisas futuras que dizem respeito a este projeto serão direcionadas nesse sentido. Tais tecnologias permitem realizar inúmeras tarefas de sumarização e trazer mais inteligência para soluções que possam ajudar no trabalho de cientistas e pesquisadores acadêmicos. Tal oportunidade abre espaço para a continuação do projeto em um programa de mestrado.

Referências Bibliográficas

- Adami, Christoph. 2021. A brief history of artificial intelligence research. *Artificial life*, **27**(2), 131–137.
- Aggarwal, Charu C, & Zhai, ChengXiang. 2012. A survey of text clustering algorithms. *Mining text data*, 77–128.
- Aisopos, Fotis, Papadakis, George, & Varvarigou, Theodora. 2011. Sentiment analysis of social media content using n-gram graphs. *Pages 9–14 of: Proceedings of the 3rd ACM SIGMM international workshop on Social media*.
- Alvares, Reinaldo Viana, Garcia, Ana Cristina Bicharra, & Ferraz, Inhaúma. 2005. STEMBR: A stemming algorithm for the Brazilian Portuguese language. *Pages 693–701 of: Portuguese conference on artificial intelligence*. Springer.
- Andrew, NG. 2016. Machine learning. *Coursera*. Available online: <https://www.coursera.org/learn/machine-learning> (accessed on 1 March 2019).
- Antoniak, Maria, & Mimno, David. 2018. Evaluating the stability of embedding-based word similarities. *Transactions of the Association for Computational Linguistics*, **6**, 107–119.
- Aranha, Christian Nunes. 2007. Uma abordagem de pré-processamento automático para mineração de textos em português: sob o enfoque da inteligência computacional. *Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ*, 33–34.
- Aria, Massimo, & Cuccurullo, Corrado. 2017. bibliometrix: An R-tool for comprehensive science mapping analysis. *Journal of informetrics*, **11**(4), 959–975.
- Armağan, Abdullah. 2013. How to write an introduction section of a scientific article? *Turkish journal of urology*, **39**(Suppl 1), 8.
- Asai, Akari, Yu, Xinyan, Kasai, Jungo, & Hajishirzi, Hanna. 2021. One question answering model for many languages with cross-lingual dense passage retrieval. *Advances in Neural Information Processing Systems*, **34**, 7547–7560.
- Asimov, Isaac. 2004. *I, robot*. Vol. 1. Spectra.

Batista, Daniel Vieira. 2025. *Summarticles*. Código-fonte. Repositório GitHub. Disponível em <https://github.com/Vieirbat/Summarticles>, acessado 04 de Janeiro, 2026.

Bellini, Valentina, Cascella, Marco, Cutugno, Franco, Russo, Michele, Lanza, Roberto, Compagnone, Christian, & Bignami, Elena. 2022. Understanding basic principles of Artificial Intelligence: a practical guide for intensivists. *Acta bio-medica : Atenei Parmensis*, **93**(10), e2022297.

Bengio, Yoshua, & Bengio, Samy. 1999. Modeling high-dimensional discrete data with multi-layer neural networks. *Advances in Neural Information Processing Systems*, **12**.

Bengio, Yoshua, Ducharme, Réjean, & Vincent, Pascal. 2000. A neural probabilistic language model. *Advances in neural information processing systems*, **13**.

Bennett, Chris, Ryall, Jody, Spalteholz, Leo, & Gooch, Amy. 2007. The aesthetics of graph visualization. *Pages 57–64 of: CAe*.

Bird, Steven, Klein, Ewan, & Loper, Edward. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. "O'Reilly Media, Inc.".

Bishop, Christopher M, & Nasrabadi, Nasser M. 2006. *Pattern recognition and machine learning*. Vol. 4. Springer.

Blei, David M, Ng, Andrew Y, & Jordan, Michael I. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, **3**(Jan), 993–1022.

Bojanowski, Piotr, Grave, Edouard, Joulin, Armand, & Mikolov, Tomas. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, **5**(06), 135–146.

Booth, Andrew Donald, Locke, William N, *et al.* 1955. *Machine translation of languages*.

Borgo, Rita, Abdul-Rahman, Alfie, Mohamed, Farhan, Grant, Philip W, Reppa, Irene, Floridi, Luciano, & Chen, Min. 2012. An empirical study on using visual embellishments in visualization. *IEEE Transactions on Visualization and Computer Graphics*, **18**(12), 2759–2768.

- Bornmann, Lutz, & Mutz, Rüdiger. 2015. Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *Journal of the association for information science and technology*, **66**(11), 2215–2222.
- Boustany, Joumana. 1997. *La production des imprimés non périodiques au Liban de 1733 à 1920: étude bibliométrique*. Ph.D. thesis, Bordeaux 3.
- Braun, Virginia, & Clarke, Victoria. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology*, **3**(01), 77–101.
- Breiman, Leo. 2001. Random forests. *Machine learning*, **45**(1), 5–32.
- Brooke, Benjamin S, Schwartz, Todd A, & Pawlik, Timothy M. 2021. MOOSE reporting guidelines for meta-analyses of observational studies. *JAMA surgery*, **156**(8), 787–788.
- Brown, Tom B. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- Brunette, Emma S, Flemmer, Rory C, & Flemmer, Claire L. 2009. A review of artificial intelligence. *Pages 385–392 of: 2009 4th International Conference on Autonomous Robots and Agents*. Ieee.
- Buckley, Chris. 1993. The importance of proper weighting methods. *In: Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.
- Burkardt, John. 2009. K-means clustering. *Virginia Tech, Advanced Research Computing, Interdisciplinary Center for Applied Mathematics*.
- Carion, Nicolas, Massa, Francisco, Synnaeve, Gabriel, Usunier, Nicolas, Kirillov, Alexander, & Zagoruyko, Sergey. 2020. End-to-End Object Detection with Transformers. *Pages 213–229 of: Vedaldi, Andrea, Bischof, Horst, Brox, Thomas, & Frahm, Jan-Michael (eds), Computer Vision – ECCV 2020*. Cham: Springer International Publishing.
- Catae, Fabricio Shigueru. 2012. *Classificação automática de texto por meio de similaridade de palavras: um algoritmo mais eficiente*. Ph.D. thesis, Universidade de São Paulo.

Cavnar, William B, Trenkle, John M, *et al.* 1994. N-gram-based text categorization. *Page 14 of: Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval*, vol. 161175. Ann Arbor, Michigan.

Cerf, Vinton G. 1969. *RFC0020: ASCII format for network interchange*.

Chan, Winnie Wing-Yi. 2006. A survey on multivariate data visualization. *Department of Computer Science and Engineering. Hong Kong University of Science and Technology*, **8**(6), 1–29.

Chintalapati, Srikrishna, & Pandey, Shivendra Kumar. 2022. Artificial intelligence in marketing: A systematic literature review. *International Journal of Market Research*, **64**(1), 38–68.

Chowdhary, KR1442. 2020. Natural language processing. *Fundamentals of artificial intelligence*, 603–649.

Church, Alonzo. 1936. A bibliography of symbolic logic. *The journal of symbolic logic*, **1**(4), 121–216.

Churchill, Rob, & Singh, Lisa. 2022. The evolution of topic modeling. *ACM Computing Surveys*, **54**(10s), 1–35.

Cobo, Manuel J, López-Herrera, Antonio Gabriel, Herrera-Viedma, Enrique, & Herrera, Francisco. 2012. SciMAT: A new science mapping analysis software tool. *Journal of the American Society for Information Science and Technology*, **63**(8), 1609–1630.

Cohen, Jérémie F, Korevaar, Daniël A, Altman, Douglas G, Bruns, David E, Gatsonis, Constantine A, Hooft, Lotty, Irwig, Les, Levine, Deborah, Reitsma, Johannes B, De Vet, Henrica CW, *et al.* 2016. STARD 2015 guidelines for reporting diagnostic accuracy studies: explanation and elaboration. *BMJ open*, **6**(11), e012799.

Cui, Mengyao, *et al.* 2020. Introduction to the k-means clustering algorithm based on the elbow method. *Accounting, Auditing and Finance*, **1**(1), 5–8.

Cukier, Kenneth. 2010. *Data, data everywhere: A special report on managing information*. Economist Newspaper.

- David, Arthur, & Sergei, Vassilvitskii. 2006. *k-means++: The advantages of careful seeding*. Tech. rept. Stanford.
- de Assis, Joaquim Maria Machado. 1899. *Dom Casmurro*. Penguin Clássicos.
- De Assis, Machado. 1998. *Memórias póstumas de Brás cubas*. Vol. 5. Ateliê Editorial.
- de Medeiros, Ivan Luiz, Vieira, Alessandro, Braviano, Gilson, & Gonçalves, Berenice Santos. 2015. Revisão Sistemática e Bibliometria facilitadas por um Canvas para visualização de informação. *InfoDesign-Revista Brasileira de Design da Informação*, **12**(1), 93–110.
- De Oliveira, MC Ferreira, & Levkowitz, Haim. 2003. From visual data exploration to visual data mining: A survey. *IEEE transactions on visualization and computer graphics*, **9**(3), 378–394.
- Deerwester, Scott, Dumais, Susan T, Furnas, George W, Landauer, Thomas K, & Harshman, Richard. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, **41**(6), 391–407.
- Deerwester, Scott C, Dumais, Susan T, Furnas, George W, Harshman, Richard A, Landauer, Thomas K, Lochbaum, Karen E, & Streeter, Lynn A. 1989 (June 13). *Computer information retrieval using latent semantic structure*. US Patent 4,839,853.
- Devlin, Jacob. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Dey, Atanu, Jenamani, Mamata, & Thakkar, Jitesh J. 2018. Senti-N-Gram: An n-gram lexicon for sentiment analysis. *Expert Systems with Applications*, **103**, 92–105.
- Dike, Happiness Ugochi, Zhou, Yimin, Deveerasetty, Kranthi Kumar, & Wu, Qingtian. 2018. Unsupervised learning based on artificial neural network: A review. *Pages 322–327 of: 2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)*. IEEE.
- Doan, AnHai, Naughton, Jeff, Baid, Akanksha, Chai, Xiaoyong, Chen, Fei, Chen, Ting, Chu, Eric, DeRose, Pedro, Gao, Byron, Gokhale, Chaitanya, *et al.* 2009. The case for a structured approach to managing unstructured data. *arXiv preprint arXiv:0909.1783*.

- Domingos, Pedro. 2015. *The master algorithm: How the quest for the ultimate learning machine will remake our world*. Basic Books.
- Dong, Guozhu, & Liu, Huan. 2018. *Feature engineering for machine learning and data analytics*. CRC press.
- Donoho, David L, *et al.* 2000. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, **1**(2000), 32.
- Dufour, Paul. 2015. Science powers commerce—but not only. *UNESCO science report: towards 2030*, 105.
- Dumais, Susan T. 1991. Improving the retrieval of information from external sources. *Behavior research methods, instruments, & computers*, **23**(2), 229–236.
- Dumais, Susan T, Furnas, George W, Landauer, Thomas K, Deerwester, Scott, & Harshman, Richard. 1988. Using latent semantic analysis to improve access to textual information. *Pages 281–285 of: Proceedings of the SIGCHI conference on Human factors in computing systems*.
- Dwivedi, Yogesh K, Hughes, Laurie, Ismagilova, Elvira, Aarts, Gert, Coombs, Crispin, Crick, Tom, Duan, Yanqing, Dwivedi, Rohita, Edwards, John, Eirug, Aled, *et al.* 2021. Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. *International Journal of Information Management*, **57**, 101994.
- Enholm, Ida Merete, Papagiannidis, Emmanouil, Mikalef, Patrick, & Krogstie, John. 2022. Artificial intelligence and business value: A literature review. *Information Systems Frontiers*, **24**(5), 1709–1734.
- Ester, Martin, Kriegel, Hans-Peter, Sander, Jörg, Xu, Xiaowei, *et al.* 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *Pages 226–231 of: kdd*, vol. 96.
- Faceli, Katti, Lorena, Ana Carolina, Gama, João, & Carvalho, André Carlos Ponce de Leon Ferreira de. 2011. *Inteligência artificial: uma abordagem de aprendizado de máquina*. LTC.

- Feldman, Ronen, & Sanger, James. 2006. Information extraction. *The text mining handbook: Advanced approaches in analyzing unstructured data*, 94–130.
- Feldman, Ronen, & Sanger, James. 2007. *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press.
- Fidelis, Joubert Roberto Ferreira, Barbosa, Ricardo Rodrigues, dos Santos, Raimundo Nonato Macedo, & Kobashi, Nair Yumiko. 2009. Bibliometria, cientometria, infometria: conceitos e aplicações. *Tendências da Pesquisa brasileira em Ciência da Informação*, **2**(1).
- Fire, Michael, & Guestrin, Carlos. 2019. Over-optimization of academic publishing metrics: observing Goodhart’s Law in action. *GigaScience*, **8**(6), giz053.
- Firth, J. 1957. A Synopsis of Linguistic Theory 1930-1955. In: *Studies in Linguistic Analysis*. Philological Society, Oxford. reprinted in Palmer, F. (ed. 1968) Selected Papers of J. R. Firth, Longman, Harlow.
- Fisher, R. A. 1936a. *Iris*. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C56C76>.
- Fisher, Ronald A. 1936b. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, **7**(2), 179–188.
- Gao, Yunfan, Xiong, Yun, Gao, Xinyu, Jia, Kangxiang, Pan, Jinliu, Bi, Yuxi, Dai, Yi, Sun, Jiawei, & Wang, Haofen. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.
- Gastel, Barbara, & Day, Robert A. 2022. *How to write and publish a scientific paper*. Bloomsbury Publishing USA.
- Gatzioufa, Paraskevi, & Saprikis, Vaggelis. 2022. A literature review on users’ behavioral intention toward chatbots’ adoption. *Applied Computing and Informatics*.
- Ghahramani, Zoubin. 2003. Unsupervised learning. Pages 72–112 of: *Summer school on machine learning*. Springer.
- Gomaa, Wael H, Fahmy, Aly A, et al. 2013. A survey of text similarity approaches. *international journal of Computer Applications*, **68**(13), 13–18.

- Gomes, Felipe Tassario, & Pardo, Thiago Alexandre Salgueiro. 2009. Classificação e agrupamento de textos para processamento multidocumento. *Universidade de São Paulo*.
- Graham, Lisa. 2008. Gestalt theory in interactive media design. *Journal of Humanities & Social Sciences*, **2**(1).
- Gross, Herbert, Blechinger, Fritz, & Achtner, Bertram. 2008. Human eye. *Handbook of Optical Systems: Volume 4: Survey of Optical Instruments*, **4**, 1–87.
- Guelpeli, Marcus Vinicius Carvalho. 2012. Cassiopeia: Um modelo de agrupamento de textos baseado em sumarização. *Niterói: Tese (Doutorado em Computação)-Universidade Federal Fluminense*.
- Gui, Jie, Chen, Tuo, Zhang, Jing, Cao, Qiong, Sun, Zhenan, Luo, Hao, & Tao, Dacheng. 2024. A Survey on Self-supervised Learning: Algorithms, Applications, and Future Trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Gunawan, Dani, Sembiring, CA, & Budiman, Mohammad Andri. 2018. The implementation of cosine similarity to calculate text relevance between two documents. *Page 012120 of: Journal of physics: conference series*, vol. 978. IOP Publishing.
- Gupta, Deepak, & Rani, Rinkle. 2019. A study of big data evolution and research challenges. *Journal of information science*, **45**(3), 322–340.
- Gutiérrez, Luis, & Keith, Brian. 2019. A systematic literature review on word embeddings. *Pages 132–141 of: Trends and Applications in Software Engineering: Proceedings of the 7th International Conference on Software Process Improvement (CIMPS 2018)* 7. Springer.
- Hänig, Christian, Schierle, Martin, & Trabold, Daniel. 2010. Comparison of structured vs. unstructured data for industrial quality analysis. *Pages 20–22 of: Proceedings of the World Congress on Engineering and Computer Science*, vol. 1.
- Hanson, Mark A, Barreiro, Pablo Gómez, Crosetto, Paolo, & Brockington, Dan. 2023. The strain on scientific publishing. *arXiv preprint arXiv:2309.15884*.
- Hardeniya, Nitin, Perkins, Jacob, Chopra, Deepti, Joshi, Nisheeth, & Mathur, Iti. 2016. *Natural language processing: python and NLTK*. Packt Publishing Ltd.

Harris, Charles R., Millman, K. Jarrod, van der Walt, Stéfan J., Gommers, Ralf, Virtanen, Pauli, Cournapeau, David, Wieser, Eric, Taylor, Julian, Berg, Sebastian, Smith, Nathaniel J., Kern, Robert, Picus, Matti, Hoyer, Stephan, van Kerkwijk, Marten H., Brett, Matthew, Haldane, Allan, del Río, Jaime Fernández, Wiebe, Mark, Peterson, Pearu, Gérard-Marchant, Pierre, Sheppard, Kevin, Reddy, Tyler, Weckesser, Warren, Abbasi, Hameer, Gohlke, Christoph, & Oliphant, Travis E. 2020. Array programming with NumPy. *Nature*, **585**(7825), 357–362.

Harris, Zellig S. 1954a. Distributional Structure. *WORD*, **10**(2-3), 146–162.

Harris, ZS. 1954b. *Distributional structure*.

Hassan, Hany, Aue, Anthony, Chen, Chang, Chowdhary, Vishal, Clark, Jonathan, Federmann, Christian, Huang, Xuedong, Junczys-Dowmunt, Marcin, Lewis, William, Li, Mu, *et al.* 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Hastie, Trevor, Tibshirani, Robert, Friedman, Jerome H, & Friedman, Jerome H. 2009. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer.

Haykin, Simon. 2001. *Redes neurais: princípios e prática*. Bookman Editora.

Hayles, N Katherine. 2010. How we read: Close, hyper, machine. *ADE bulletin*, **150**(18), 62–79.

Haynes, Alex B, Haukoos, Jason S, & Dimick, Justin B. 2021. TREND reporting guidelines for nonrandomized/quasi-experimental study designs. *JAMA surgery*, **156**(9), 879–880.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, & Sun, Jian. 2015. *Deep Residual Learning for Image Recognition*.

Hearst, Marti A, Pedersen, Emily, Patil, Lekha, Lee, Elsie, Laskowski, Paul, & Franconeri, Steven. 2019. An evaluation of semantically grouped word cloud designs. *IEEE transactions on visualization and computer graphics*, **26**(9), 2748–2761.

Henry, ER, & Hofrichter, J. 1992. Singular value decomposition: Application to analysis of experimental data. *Pages 129–192 of: Methods in enzymology*, vol. 210. Elsevier.

- Hullman, Jessica, Adar, Eytan, & Shah, Priti. 2011. Benefitting infovis with visual difficulties. *IEEE Transactions on Visualization and Computer Graphics*, **17**(12), 2213–2222.
- Ibnu, Choldun R Muh, Santoso, Judhi, & Surendro, Kridanto. 2019. Determining the neural network topology: A review. *Pages 357–362 of: Proceedings of the 2019 8th International Conference on Software and Computer Applications*.
- ICMJE, International Committee of Medical Journal Editors, *et al.* 2006. ICMJE, Uniform requirements for manuscripts submitted to biomedical journals: Writing and editing for biomedical publication International Committee of Medical Journal Editors Updated October 2005 (www.icmje.org). *Indian Journal of Pharmacology*, **38**(2), 149.
- Islam, Aminul, & Inkpen, Diana. 2008. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, **2**(2), 1–25.
- Jain, Anil K, & Dubes, Richard C. 1988. *Algorithms for clustering data*. Prentice-Hall, Inc.
- James, Gareth, Witten, Daniela, Hastie, Trevor, & Tibshirani, Robert. 2013. *An introduction to statistical learning*. Vol. 112. Springer.
- Jara Casco, Eugenio. 1999. La selección del título en el artículo científico. *Revista Cubana de Medicina General Integral*, **15**, 342–345.
- Jiang, Jian, Chen, Long, Ke, Lu, Dou, Bozheng, Zhang, Chunhuan, Feng, Hongsong, Zhu, Yueying, Qiu, Huahai, Zhang, Bengong, & Wei, Guowei. 2024. A review of transformers in drug discovery and beyond. *Journal of Pharmaceutical Analysis*, 101081.
- Jinha, Arif. 2010. Article 50 million: An estimate of the number of scholarly articles in existence. *Learned Publishing*, **23**(07), 258–263.
- Jones, Karen Sparck. 1994. Natural language processing: a historical review. *Current issues in computational linguistics: in honour of Don Walker*, 3–16.
- Joos, Martin. 1950. Description of language design. *The Journal of the Acoustical Society of America*, **22**(6), 701–707.

Joseilme Fernandes Gouveia, Maria Luzitana Conceição dos Santos. 2017. *MANUAL PARA A ELABORAÇÃO DE ARTIGO CIENTÍFICO (TRABALHO DE CONCLUSÃO DE CURSO).*

Josko, João Marcelo Borovina. 2023. *Aula 1 - Sistema Visual Humano: O Processo Biológico, Slide 5.* Apresentação de aula em slides, disciplina MCZA052-22 Vizualização de Dados e Informações.

JSAI, Japanese Society for Artificial Intelligence (JSAI), (JSAI). 2021. *AI Map beta 2.0.*

Jurafsky, Daniel, & Martin, James H. 2024. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models.* 3rd edn. Pearson Prentice Hall, Upper Saddle River, New Jersey 07458. Online manuscript released August 20, 2024.

Kaisler, Stephen H. 2016. *Birthing the computer: from relays to vacuum tubes.* Cambridge Scholars Publishing.

Kapil, Shruti, & Chawla, Meenu. 2016. Performance evaluation of K-means clustering algorithm with various distance metrics. *Pages 1–4 of: 2016 IEEE 1st international conference on power electronics, intelligent control and energy systems (ICPEICES).* IEEE.

Kashyap, Abhinav Ramesh, & Kan, Min-Yen. 2020. *SciWING – A Software Toolkit for Scientific Document Processing.*

Kaufmann, Timo, Weng, Paul, Bengs, Viktor, & Hüllermeier, Eyke. 2023. A survey of reinforcement learning from human feedback. *arXiv preprint arXiv:2312.14925.*

Keim, Daniel A. 1997. Visual Techniques for Exploring Databases. *In: Knowledge Discovery and Data Mining.*

Keim, Daniel A, & Kriegel, H-P. 1996. Visualization techniques for mining large databases: A comparison. *IEEE Transactions on knowledge and data engineering*, 8(6), 923–938.

Kenton, Jacob Devlin Ming-Wei Chang, & Toutanova, Lee Kristina. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Page 2 of: Proceedings of naacl-HLT*, vol. 1. Minneapolis, Minnesota.

Keshav, Srinivasan. 2007. How to read a paper. *ACM SIGCOMM Computer Communication Review*, **37**(3), 83–84.

Khan, Muzammil, & Khan, Sarwar Shah. 2011. Data and information visualization methods, and interactive mechanisms: A survey. *International Journal of Computer Applications*, **34**(1), 1–14.

Khan, Salman, Naseer, Muzammal, Hayat, Munawar, Zamir, Syed Waqas, Khan, Fahad Shah-baz, & Shah, Mubarak. 2022. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, **54**(10s), 1–41.

Kherwa, Pooja, & Bansal, Poonam. 2019. Topic modeling: a comprehensive review. *EAI Endorsed transactions on scalable information systems*, **7**(24).

Khokhar, Devangana. 2015. *Gephi cookbook*. Packt Publishing Ltd.

Kobayashi, Mei, & Takeda, Koichi. 2000. Information retrieval on the web. *ACM computing surveys (CSUR)*, **32**(2), 144–173.

Kobourov, Stephen G, Mchedlidze, Tamara, & Vonessen, Laura. 2015. Gestalt principles in graph drawing. *Pages 558–560 of: Graph Drawing and Network Visualization: 23rd International Symposium, GD 2015, Los Angeles, CA, USA, September 24-26, 2015, Revised Selected Papers 23*. Springer.

Kondrak, Grzegorz. 2005. N-gram similarity and distance. *Pages 115–126 of: International symposium on string processing and information retrieval*. Springer.

Koopman, Philip. 1997. *How to write an abstract*.

Köppen, Mario. 2000. The curse of dimensionality. *Pages 4–8 of: 5th online world conference on soft computing in industrial applications (WSC5)*, vol. 1.

Krizhevsky, Alex, Sutskever, Ilya, & Hinton, Geoffrey E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, **25**.

- Kumar, Sunil, Gupta, Rajat, Khanna, Nitin, Chaudhury, Santanu, & Joshi, Shiv Dutt. 2007. Text extraction and document image segmentation using matched wavelets and MRF model. *IEEE Transactions on Image Processing*, **16**(8), 2117–2128.
- Lancaster, Frederic Wilfrid. 2004. Indexação e resumos. *Tradução de AA Briquet de Lemos. Brasilia.*
- Landauer, Thomas K, Laham, Darrell, Rehder, Bob, & Schreiner, Missy E. 1997. How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans. *Pages 412–417 of: Proceedings of the 19th annual meeting of the Cognitive Science Society.*
- Landhuis, Esther. 2016. Scientific literature: Information overload. *Nature*, **535**(7612), 457–458.
- Latif, Siddique, Zaidi, Aun, Cuayahuitl, Heriberto, Shamshad, Fahad, Shoukat, Moazzam, & Qadir, Junaid. 2023. Transformers in speech processing: A survey. *arXiv preprint arXiv:2303.11607.*
- Lawani, Stephen Majebi. 1981. Bibliometrics: Its theoretical foundations, methods and applications. *Libri*, **31**(Jahresband), 294–315.
- Lewis, Patrick, Perez, Ethan, Piktus, Aleksandra, Petroni, Fabio, Karpukhin, Vladimir, Goyal, Naman, Küttler, Heinrich, Lewis, Mike, Yih, Wen-tau, Rocktäschel, Tim, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, **33**, 9459–9474.
- Leximancer Pty Ltd. 2005-2026. *Leximancer*. Disponível em <https://www.leximancer.com/>. Acessado 16 de Abril 2025.
- Li, Wenzhe, Luo, Hao, Lin, Zichuan, Zhang, Chongjie, Lu, Zongqing, & Ye, Deheng. 2023. A survey on transformers in reinforcement learning. *arXiv preprint arXiv:2301.03044.*
- Lichtfouse, Eric. 2013. Scientific writing for impact factor journals. *HAL SHS Nova Publishers.*

Limiro, Renata Moreira, Da Silva, Núbia Rosa, & Cordeiro, Douglas Farias. 2022. Mineração de textos para agrupamento de teses e dissertações por meio de análise de similaridade. *Revista Brasileira de Biblioteconomia e Documentação*, **18**, 1–20.

Lin, Chin-Yew, & Hovy, Eduard. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. *Pages 150–157 of: Proceedings of the 2003 human language technology conference of the North American chapter of the association for computational linguistics*.

Lin, Tianyang, Wang, Yuxin, Liu, Xiangyang, & Qiu, Xipeng. 2022. A survey of transformers. *AI Open*, **3**, 111–132.

Lloyd, Stuart. 1982. Least squares quantization in PCM. *IEEE transactions on information theory*, **28**(2), 129–137.

Locke, John L, & Bogin, Barry. 2006. Language and life history: A new perspective on the development and evolution of human language. *Behavioral and Brain Sciences*, **29**(3), 259–280.

Lopez, Patrice. 2008–2025. *GROBID*. <https://github.com/kermitt2/grobid>. Versão 0.8.0. Acesso em 04 jan. 2026.

Lovins, Julie Beth. 1968. Development of a stemming algorithm. *Mech. Transl. Comput. Linguistics*, **11**(1-2), 22–31.

Luhn, Hans Peter. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, **2**(2), 159–165.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. *In: Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*.

Mahmood, ASMA. 2009. Literature survey on topic modeling. *Technical report*.

Manning, Christopher D. 1999. *Foundations of statistical natural language processing*. The MIT Press.

Marcos, & Souza, Renato Rocha. 2019. Modelagem de tópicos: Resumir e organizar corpus de dados por meio de algoritmos de aprendizagem de máquina. *Múltiplos Olhares em Ciência da Informação*, **9**(2).

Marcus, Gary. 2020. The next decade in AI: four steps towards robust artificial intelligence. *arXiv preprint arXiv:2002.06177*.

Matsubara, Edson Takashi, Martins, Claudia Aparecida, & Monard, Maria Carolina. 2003. *PreText: uma ferramenta para pré-processamento de textos utilizando a abordagem bag-of-words*.

McCarthy, J, Minsky, ML, & Rochester, N. 1956. The Dartmouth summer research project on artificial intelligence. Artificial intelligence: past, present, and future. <https://www.aaai.org/ojs/index.php/aimagazine/article/view/1904/1802>.

McCulloch, Warren S, & Pitts, Walter. 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, **5**, 115–133.

Meadows, Arthur Jack, & de Lemos Lemos, Antonio Agenor Briquet. 1999. *A comunicação científica*. Briquet de Lemos/livros.

Merkel, Dirk. 2014. Docker: lightweight linux containers for consistent development and deployment. *Linux Journal*, **2014**(239), 2.

Mialon, Grégoire, Dessì, Roberto, Lomeli, Maria, Nalmpantis, Christoforos, Pasunuru, Ram, Raileanu, Roberta, Rozière, Baptiste, Schick, Timo, Dwivedi-Yu, Jane, Celikyilmaz, Asli, et al. 2023. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*.

Mikolov, Tomas. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg S, & Dean, Jeff. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, **26**.

- Minaee, Shervin, Mikolov, Tomas, Nikzad, Narjes, Chenaghlu, Meysam, Socher, Richard, Amatriain, Xavier, & Gao, Jianfeng. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.
- Mitchell, Tom Michael, *et al.* 1997. *Machine learning*. Vol. 1. McGraw-hill New York.
- Moere, Andrew Vande, Tomitsch, Martin, Wimmer, Christoph, Christoph, Boesch, & Grechenig, Thomas. 2012. Evaluating the effect of style in information visualization. *IEEE transactions on visualization and computer graphics*, **18**(12), 2739–2748.
- Monmonier, Mark. 1985. *Semiology of Graphics: Diagrams, Networks, Maps*.
- Morais, Edison Andrade Martins, & Ambrósio, Ana Paula L. 2007. Mineração de textos. *Relatório Técnico–Instituto de Informática (UFG)*.
- Mueller, Suzana Pinheiro Machado. 1995. O crescimento da ciência, o comportamento científico e a comunicação científica: algumas reflexões. *Revista da Escola de Biblioteconomia da UFMG*, **24**(1).
- Muhammad, Iqbal, & Yan, Zhu. 2015. SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY. *ICTACT Journal on Soft Computing*, **5**(3).
- Nerella, Subhash, Bandyopadhyay, Sabyasachi, Zhang, Jiaqing, Contreras, Miguel, Siegel, Scott, Bumin, Aysegul, Silva, Brandon, Sena, Jessica, Shickel, Benjamin, Bihorac, Azra, *et al.* 2023. Transformers in healthcare: A survey. *arXiv preprint arXiv:2307.00067*.
- Nessa, Syeda, Abedin, Muhammad, Wong, W Eric, Khan, Latifur, & Qi, Yu. 2008. Software fault localization using n-gram analysis. *Pages 548–559 of: Wireless Algorithms, Systems, and Applications: Third International Conference, WASA 2008, Dallas, TX, USA, October 26–28, 2008. Proceedings* 3. Springer.
- Oke, Sunday Ayoola. 2008. A literature review on artificial intelligence. *International journal of information and management sciences*, **19**(4), 535–570.
- Oliveira, Eduardo A, Oliveira, Maria Christina L, Colosimo, Enrico A, Martelli, Daniella B, Silva, Ludmila R, Silva, Ana Cristina Simões E, & Martelli-Júnior, Hercílio. 2022. Global

- scientific production in the pre-Covid-19 Era: An analysis of 53 countries for 22 years. *Anais da Academia Brasileira de Ciências*, **94**(suppl 3), e20201428.
- Ollama. 2023. *Ollama: Get up and running with large language models*. Ollama. Disponível em <https://ollama.com/>, acessado em 04 jan. 2026.
- O'Regan. 2008. *A brief history of computing*. Springer.
- Orrego, V.M., & Huyck, C. 2001. A stemming algorithm for the portuguese language. *Pages 186–193 of: Proceedings Eighth Symposium on String Processing and Information Retrieval*.
- Osgood, Charles E., Suci, George J., & Tannenbaum, Percy H. 1957. *The Dimensionality of the Semantic Space*. Urbana, IL: University of Illinois Press. Chap. 2, pages 31–75.
- Otlet, Paul. 1934. *Traité de documentation: le livre sur le livre, theéorie et pratique*. Editiones mundaneum.
- O'Regan, Gerard. 2018. Eliza Program. *The Innovation in Computing Companion: A Compendium of Select, Pivotal Inventions*, 119–122.
- Paice, Chris D. 1990. Another stemmer. *Pages 56–61 of: ACM Sigir Forum*, vol. 24. ACM New York, NY, USA.
- Paiva, Eduardo, Paim, Andréa, & Ebecken, Nelson. 2021. Convolutional neural networks and long short-term memory networks for textual classification of information access requests. *IEEE Latin America Transactions*, **19**(5), 826–833.
- pandas development team, The. 2020 (Feb.). *pandas-dev/pandas: Pandas*.
- Papadimitriou, Christos H, Tamaki, Hisao, Raghavan, Prabhakar, & Vempala, Santosh. 1998. Latent semantic indexing: A probabilistic analysis. *Pages 159–168 of: Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*.
- Park, Jun-U, Ko, Sang-Ki, Cognetta, Marco, & Han, Yo-Sub. 2019. Softregex: Generating regex from natural language descriptions using softened regex equivalence. *Pages*

6425–6431 of: *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*.

Pasquarelli, Maria Luiza Rigo. 2004. Normas para a apresentação de trabalhos acadêmicos (ABNT/NBR-14724, AGOSTO 2002). *Edifieo. 2a edição. São Paulo: Osasco.*

Pauls, Adam, & Klein, Dan. 2011. Faster and smaller n-gram language models. *Pages 258–267 of: Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human Language Technologies.*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830.

Peng, Fuchun, & Schuurmans, Dale. 2003. Combining naive Bayes and n-gram language models for text classification. *Pages 335–350 of: European Conference on Information Retrieval*. Springer.

Pennington, Jeffrey, Socher, Richard, & Manning, Christopher D. 2014. Glove: Global vectors for word representation. *Pages 1532–1543 of: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*.

Pereira, Mauricio Gomes. 2012. A introdução de um artigo científico. *Epidemiologia e Serviços de Saúde*, **21**(4), 675–676.

Perianes-Rodriguez, Antonio, Waltman, Ludo, & Van Eck, Nees Jan. 2016. Constructing bibliometric networks: A comparison between full and fractional counting. *Journal of informetrics*, **10**(4), 1178–1195.

Pernice, Kara, Whitenton, Kathryn, & Nielsen, Jakob. 2010. *How People Read on the Web: The Eyetracking Evidence*. Nielsen Norman Group.

Piantadosi, Steven T. 2014. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, **21**, 1112–1130.

- Pilehvar, Mohammad Taher, & Camacho-Collados, Jose. 2020. *Embeddings in natural language processing: Theory and advances in vector representations of meaning*. Morgan & Claypool Publishers.
- Pinker, Steven. 2003. *The language instinct: How the mind creates language*. Penguin uK.
- PK, FATHIMA ANJILA. 1984. What is Artificial Intelligence? “*Success is no accident. It is hard work, perseverance, learning, studying, sacrifice and most of all, love of what you are doing or learning to do*”,., 65.
- Popescu, Marius-Constantin, Balas, Valentina E, Perescu-Popescu, Liliana, & Mastorakis, Nikos. 2009. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, **8**(7), 579–588.
- Porter, Martin F. 1980. An algorithm for suffix stripping. *Program*, **14**(3), 130–137.
- Pradhan, Nitesh, Gyanchandani, Manasi, Wadhvani, Rajesh, et al. 2015. A Review on Text Similarity Technique used in IR and its Application. *International Journal of Computer Applications*, **120**(9), 29–34.
- Price, DJ de S. 1969. The structures of publication in science and technology. *Factors in the Transfer of Technology*, 91–104.
- Qader, Wisam A, Ameen, Musa M, & Ahmed, Bilal I. 2019. An overview of bag of words; importance, implementation, applications, and challenges. *Pages 200–204 of: 2019 international engineering conference (IEC)*. IEEE.
- Qin, Xuedi, Luo, Yuyu, Tang, Nan, & Li, Guoliang. 2020. Making data visualization more efficient and effective: a survey. *The VLDB Journal*, **29**(1), 93–117.
- Radford, Alec, & Narasimhan, Karthik. 2018. Improving Language Understanding by Generative Pre-Training.
- Raiaan, Mohaimenul Azam Khan, Mukta, Md Saddam Hossain, Fatema, Kaniz, Fahad, Nur Mohammad, Sakib, Sadman, Mim, Most Marufatul Jannat, Ahmad, Jubaer, Ali, Mohammed Eunus, & Azam, Sami. 2024. A review on large Language Models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*.

- Rasamoelina, Andrinandrasana David, Adjailia, Fouzia, & Sinčák, Peter. 2020. A review of activation function for artificial neural network. *Pages 281–286 of: 2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE.
- Reddy, D Krishna Sandeep, & Pujari, Arun K. 2006. N-gram analysis for computer virus detection. *Journal in computer virology*, **2**, 231–239.
- Ribeiro, Alcides José Araújo. 2016. *Artigos científicos—como redigir, publicar e avaliar*.
- Rodrigues, Jose F, Traina, Agma JM, de Oliveira, Maria Cristina Ferreira, & Traina, C. 2006. Reviewing data visualization: an analytical taxonomical study. *Pages 713–720 of: Tenth International Conference on Information Visualisation (IV'06)*. IEEE.
- Rojas, Raul, & Rojas, Raúl. 1996. The backpropagation algorithm. *Neural networks: a systematic introduction*, 149–182.
- Rosenblatt, Frank. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, **65**(6), 386.
- Rostaing, Corinne. 1996. *La relation carcérale: Identités et rapports sociaux dans les prisons de femmes*. FeniXX.
- Russell, Bertrand, & Whitehead, Alfred North. 1910. *Principia Mathematica Vol. I*. Cambridge University Press.
- Russell, Stuart J, & Norvig, Peter. 2016. *Artificial intelligence: a modern approach*. Pearson.
- Sagiroglu, Seref, & Sinanc, Duygu. 2013. Big data: A review. *Pages 42–47 of: 2013 international conference on collaboration technologies and systems (CTS)*. IEEE.
- Salton, Gerard. 1983. Introduction to modern information retrieval. *McGrawHill Book Co.*
- Salton, Gerard, Wong, Anita, & Yang, Chung-Shu. 1975. A vector space model for automatic indexing. *Communications of the ACM*, **18**(11), 613–620.
- Santana, Nonilton Alves de. 2017. *A utilização da técnica de filtragem de conteúdo em campos de texto livre para recomendações de diagnóstico*. Ph.D. thesis, UEMA.

- Schiappa, Madeline C, Rawat, Yogesh S, & Shah, Mubarak. 2023. Self-supervised learning for videos: A survey. *ACM Computing Surveys*, **55**(13s), 1–37.
- Scholarcy. 2026. *Scholarcy: Knowledge made simple*. Ferramenta de sumarização de artigos baseada em IA, disponível em <https://www.scholarcy.com/>. Acessada 05 de Julho, 2025.
- Schulz, Kenneth F, Moher, David, & Altman, Douglas G. 2014. Consort. *Guidelines for Reporting Health Research: A User's Manual*, 80–92.
- Schütze, Hinrich, Manning, Christopher D, & Raghavan, Prabhakar. 2008. *Introduction to information retrieval*. Vol. 39. Cambridge University Press Cambridge.
- Sebastiani, Fabrizio. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, **34**(1), 1–47.
- Selçuk, Ayşe Adin. 2019. A guide for systematic reviews: PRISMA. *Turkish archives of otorhinolaryngology*, **57**(1), 57.
- Shinde, Pramila P, & Shah, Seema. 2018. A review of machine learning and deep learning applications. *Pages 1–6 of: 2018 Fourth international conference on computing communication control and automation (ICCUBEA)*. IEEE.
- Shneiderman, Ben. 2003. The eyes have it: A task by data type taxonomy for information visualizations. *Pages 364–371 of: The craft of information visualization*. Elsevier.
- Siivola, Vesa, & Pellom, Bryan L. 2005. Growing an n-gram language model. *Pages 1309–1312 of: Interspeech*.
- Singh, Shashi Pal, Kumar, Ajai, Darbari, Hemant, Singh, Lenali, Rastogi, Anshika, & Jain, Shikha. 2017. Machine translation using deep learning: An overview. *Pages 162–167 of: 2017 international conference on computer, communications and electronics (comptelix)*. IEEE.
- Smith, Agnese, et al. 2015. Artificial intelligence. *National*. URL: <https://nationalmagazine.ca/en-ca/articles/law/indepth/2020/artificial-intelligence> (date of access: 14.11. 2021).
- Sorzano, Carlos Oscar Sánchez, Vargas, Javier, & Montano, A Pascual. 2014. A survey of dimensionality reduction techniques. *arXiv preprint arXiv:1403.2877*.

- Srivastava, Ashok N, & Sahami, Mehran. 2009. *Text mining: Classification, clustering, and applications*. CRC press.
- Stanford Online. 2021. *Stanford CS224N: NLP with Deep Learning — Winter 2021 — Lecture 1 - Intro & Word Vectors*. Vídeo. YouTube. Publicado em: 29 out. 2021. Duração: 1 h 24 min 27 s.
- Steyvers, Mark, & Griffiths, Tom. 2007. Probabilistic topic models. *Pages 439–460 of: Handbook of latent semantic analysis*. Psychology Press.
- Streamlit Inc. 2026. *Streamlit*. <https://streamlit.io>. Versão 1.41.0. Acesso em: 04 jan. 2026.
- Suen, Ching Y. 1979. N-gram statistics for natural language understanding and text processing. *IEEE transactions on pattern analysis and machine intelligence*, 164–172.
- Sultan, Afnan, Sieg, Jochen, Mathea, Miriam, & Volkamer, Andrea. 2024. Transformers for molecular property prediction: Lessons learned from the past five years. *Journal of Chemical Information and Modeling*, 64(16), 6259–6280.
- Sutskever, I. 2014. Sequence to Sequence Learning with Neural Networks. *arXiv preprint arXiv:1409.3215*.
- Swarndep Saket, J, & Pandya, Sharnil. 2016. An overview of partitioning algorithms in clustering techniques. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 5(6), 1943–1946.
- Switzer, Paul. 1965. Vector images in document retrieval. *Statistical association methods for mechanized documentation*, 163–171.
- Synnestvedt, Marie B, Chen, Chaomei, & Holmes, John H. 2005. CiteSpace II: visualization and knowledge discovery in bibliographic databases. *Page 724 of: AMIA annual symposium proceedings*, vol. 2005. American Medical Informatics Association.
- Thanaki, Jalaj. 2017. *Python natural language processing*. Packt Publishing Ltd.
- Toosi, Amirhosein, Bottino, Andrea G, Saboury, Babak, Siegel, Eliot, & Rahmim, Arman. 2021. A brief history of AI: how to prevent another winter (a critical review). *PET clinics*, 16(4), 449–469.

- Tovée, Martin J. 2008. *An introduction to the visual system*. Cambridge University Press.
- Tripathy, Abinash, Agrawal, Ankit, & Rath, Santanu Kumar. 2016. Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, **57**, 117–126.
- Tufte, Edward R, & Graves-Morris, Peter R. 1983. *The visual display of quantitative information*. Vol. 2. Graphics press Cheshire, CT.
- Turing, Alan. 1936. Turing machine. *Proc London Math Soc*, **242**, 230–265.
- Turing, Alan M. 1950. *Computing machinery and intelligence*. Springer.
- van Eck, Nees Jan, & Waltman, Ludo. 2010. Software survey: VOSviewer, a computer program for bibliometric mapping. *Scientometrics*, **84**(2), 523–538.
- Van Eck, Nees Jan, & Waltman, Ludo. 2013. VOSviewer manual. *Leiden: Univeristeit Leiden*, **1**(1), 1–53.
- Van Engelen, Jesper E, & Hoos, Holger H. 2020. A survey on semi-supervised learning. *Machine learning*, **109**(2), 373–440.
- Van Rossum, Guido, & Drake, Fred L. 1989. *The Python Language Reference*. Python Software Foundation. Acessado em 04 jan. 2026.
- Van Wijk, Jarke J. 2005. The value of visualization. *Pages 79–86 of: VIS 05. IEEE Visualization, 2005*. IEEE.
- Vargas, José Israel. 1997. *Mecanismos de transferência de tecnologia para países do terceiro mundo*. Vol. 24. Universidade de São Paulo, Instituto de Estudos Avançados.
- Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N., Kaiser, Lukasz, & Polosukhin, Illia. 2023. *Attention Is All You Need*.
- Venkatesh, B, & Anuradha, J. 2019. A review of feature selection and its methods. *Cybernetics and information technologies*, **19**(1), 3–26.
- Vogels, Tim P, Rajan, Kanaka, & Abbott, Larry F. 2005. Neural network dynamics. *Annu. Rev. Neurosci.*, **28**(1), 357–376.

- Von Elm, Erik, Altman, Douglas G, Egger, Matthias, Pocock, Stuart J, Gøtzsche, Peter C, & Vandenbroucke, Jan P. 2007. The Strengthening the Reporting of Observational Studies in Epidemiology (STROBE) statement: guidelines for reporting observational studies. *The lancet*, **370**(9596), 1453–1457.
- von Engelhardt, Jörg. 2002. *The language of graphics: A framework for the analysis of syntax and meaning in maps, charts and diagrams*. Yuri Engelhardt.
- Voos, Henry. 1974. Lotka and information science. *Journal of the American Society for Information Science*, **25**(4), 270–272.
- Wainer, Jacques, et al. 2007. Métodos de pesquisa quantitativa e qualitativa para a Ciência da Computação. *Atualização em informática*, **1**(221-262), 32–33.
- Wang, Jiapeng, & Dong, Yihong. 2020. Measurement of text similarity: a survey. *Information*, **11**(9), 421.
- Wang, Xu, Wang, Sen, Liang, Xingxing, Zhao, Dawei, Huang, Jincai, Xu, Xin, Dai, Bin, & Miao, Qiguang. 2022. Deep reinforcement learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, **35**(4), 5064–5078.
- Webster, Jonathan J, & Kit, Chunyu. 1992. Tokenization as the initial phase in NLP. In: *COLING 1992 volume 4: The 14th international conference on computational linguistics*.
- Wegmann, Marc, Zipperling, Domenique, Hillenbrand, Jonas, & Fleischer, Jürgen. 2021. A review of systematic selection of clustering algorithms and their evaluation. *arXiv preprint arXiv:2106.12792*.
- Weiss, Karl, Khoshgoftaar, Taghi M, & Wang, DingDing. 2016. A survey of transfer learning. *Journal of Big data*, **3**, 1–40.
- Wen, Qingsong, Zhou, Tian, Zhang, Chaoli, Chen, Weiqi, Ma, Ziqing, Yan, Junchi, & Sun, Liang. 2022. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*.
- Wetzel, Linda. 2018. Types and Tokens. In: Zalta, Edward N. (ed), *The Stanford Encyclopedia of Philosophy*, Fall 2018 edn. Metaphysics Research Lab, Stanford University.

- Wilamowski, Bogdan M. 2009. Neural network architectures and learning algorithms. *IEEE Industrial Electronics Magazine*, **3**(4), 56–63.
- Wildgen, Wolfgang. 2004. *The evolution of human language*. John Benjamins Publishing Company.
- Winston, Patrick Henry. 1992. *Artificial intelligence*. Addison-Wesley Longman Publishing Co., Inc.
- Wives, Leandro Krug. 2002. Tecnologias de descoberta de conhecimento em textos aplicadas à inteligência competitiva. *Exame de Qualificação EQ-069, PPGC-UFRGS*, **18**.
- Wong, Ruth, Allen, Frank H, & Willett, Peter. 2010. The scientific impact of the Cambridge Structural Database: A citation-based study. *Journal of Applied Crystallography*, **43**(4), 811–824.
- Woschank, Manuel, Rauch, Erwin, & Zsifkovits, Helmut. 2020. A review of further directions for artificial intelligence, machine learning, and deep learning in smart logistics. *Sustainability*, **12**(9), 3760.
- Xu, Dongkuan, & Tian, Yingjie. 2015. A comprehensive survey of clustering algorithms. *Annals of data science*, **2**, 165–193.
- Yadav, Ashima, & Vishwakarma, Dinesh Kumar. 2020. Sentiment analysis using deep learning architectures: a review. *Artificial Intelligence Review*, **53**(6), 4335–4385.
- Yang, Jingfeng, Jin, Hongye, Tang, Ruixiang, Han, Xiaotian, Feng, Qizhang, Jiang, Haoming, Zhong, Shaochen, Yin, Bing, & Hu, Xia. 2024. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, **18**(6), 1–32.
- Yu, Kun-Hsing, Beam, Andrew L, & Kohane, Isaac S. 2018. Artificial intelligence in health-care. *Nature biomedical engineering*, **2**(10), 719–731.
- Zhao, Wayne Xin, Zhou, Kun, Li, Junyi, Tang, Tianyi, Wang, Xiaolei, Hou, Yupeng, Min, Yingqian, Zhang, Beichen, Zhang, Junjie, Dong, Zican, *et al.* 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zhuang, Fuzhen, Qi, Zhiyuan, Duan, Keyu, Xi, Dongbo, Zhu, Yongchun, Zhu, Hengshu, Xiong, Hui, & He, Qing. 2020. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, **109**(1), 43–76.

Zipf, George Kingsley. 2016. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio books.

Zouhar, Vilém, Meister, Clara, Gastaldi, Juan Luis, Du, Li, Vieira, Tim, Sachan, Mrinmaya, & Cotterell, Ryan. 2023. A formal perspective on byte-pair encoding. *arXiv preprint arXiv:2306.16837*.