FUNDAÇÃO EDSON QUEIROZ
UNIVERSIDADE DE FORTALEZA - UNIFOR
CENTRO DE CIÊNCIAS TECNOLÓGICAS
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

AUTOMAÇÃO DO PROCESSO DE CONFECÇÃO DE ATAS DE REUNIÃO COM LLM'S NO CONTEXTO JURÍDICO

HENRIQUE FAÇANHA DUTRA

FORTALEZA – CEARÁ
2025

HENRIQUE FAÇANHA DUTRA

AUTOMAÇÃO DO PROCESSO DE CONFECÇÃO DE ATAS DE REUNIÃO COM LLM'S NO CONTEXTO JURÍDICO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação do Centro de Ciências Tecnológicas da Universidade de Fortaleza, como requisito parcial à obtenção do grau de bacharel em Ciência da Computação.

Orientador: Juliana Martins de Oliveira

FORTALEZA – CEARÁ

2025

A ficha catalográfica deve ser gerada no site da biblioteca da Unifor através do link https://goo.gl/XYUWSC (link encurtado).

Preencha o formulário com as informações solicitadas e ao final será gerado um arquivo PDF da ficha catalográfica a ser anexada na versão final do TCC.

O arquivo PDF deve ser renomeado para "ficha-catalografica.pdf" (sem aspas) e colocado no diretório "elementos-pre-textuais" (sem aspas) do modelo de TCC da Unifor.

Ficha catalográfica da obra elaborada pelo autor através do programa de geração automática da Biblioteca Central da Universidade de Fortaleza

---

Batista, Bruno .
TEORIA DA RELATIVIDADE: SUBTÍTULO / Bruno Batista, Sandra Lima. - 2017
40 f.

Trabalho de Conclusão de Curso (Graduação) - Universidade de Fortaleza. Curso de Ciência da Computação, Fortaleza, 2017.
Orientação: Liadina Camargo.

1. FÍSICA. 2. RELATIVIDADE. 3. TEMPO. 4. ESPAÇO. I. Lima, Sandra. II. Camargo, Liadina. III. Título.

ERRATA

HENRIQUE FAÇANHA DUTRA

AUTOMAÇÃO DO PROCESSO DE CONFECÇÃO DE ATAS DE REUNIÃO COM LLM'S NO CONTEXTO JURÍDICO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação do Centro de Ciências Tecnológicas da Universidade de Fortaleza, como requisito parcial à obtenção do grau de bacharel em Ciência da Computação.

Aprovada em:

BANCA EXAMINADORA

Juliana Martins de Oliveira (Orientador)
Centro de Ciências Tecnológicas - CCT
Universidade de Fortaleza - UNIFOR

Membro da Banca Dois
Centro de Ciências e Tecnologia - CCT
Universidade do Membro da Banca Dois - SIGLA

Membro da Banca Três
Centro de Ciências e Tecnologia - CCT
Universidade do Membro da Banca Três - SIGLA

Membro da Banca Quatro
Centro de Ciências e Tecnologia - CCT
Universidade do Membro da Banca Quatro - SIGLA

Este trabalho é dedicado às crianças adultas que, quando pequenas, sonharam em se tornar cientistas.

AGRADECIMENTOS

Primeiramente a Deus que permitiu que tudo isso acontecesse, ao longo de minha vida, e não somente nestes anos como universitária, mas que em todos os momentos é o maior mestre que alguém pode conhecer.

Aos meus pais, pelo amor, incentivo e apoio incondicional.

Obrigada meus irmãos e sobrinhos, que nos momentos de minha ausência dedicados ao estudo superior, sempre fizeram entender que o futuro é feito a partir da constante dedicação no presente!

A esta universidade, seu corpo docente, direção e administração que oportunizaram a janela que hoje vislumbro um horizonte superior, eivado pela acendrada confiança no mérito e ética aqui presentes.

Agradeço a todos os professores por me proporcionar o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender. a palavra mestre, nunca fará justiça aos professores dedicados aos quais sem nominar terão os meus eternos agradecimentos.

"É melhor lançar-se à luta em busca do triunfo mesmo expondo-se ao insucesso, que formar fila com os pobres de espírito, que nem gozam muito nem sofrem muito; E vivem nessa penumbra cinzenta sem conhecer nem vitoria nem derrota."

(Franklin Roosevelt)

RESUMO

Palavras-chave: LLM. IA. MCP. transformer

ABSTRACT

Keywords: LLM. IA. MCP. transformer

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE ALGORITMOS

LISTA DE CÓDIGOS-FONTE

LISTA DE SÍMBOLOS

AM	Aprendizado de Máquina
GPU	Unidade de Processamento Gráfico (Graphics Processing Unit)
GRU	Unidades Recorrentes com Portões (Gated Recurrent Units)
IA	Inteligência Artificial
LLM	Modelo de Linguagem de Grande Porte (Large Language Model)
LSTM	Memória de Longo Curto Prazo (Long Short-Term Memory)
PLM	Modelos de Linguagem Pré-treinados (Pre-trained Language Models)
PLN	Processamento de Linguagem Natural
RNA	Redes Neurais Artificiais
RNN	Redes Neurais Recorrentes (Recurrent Neural Networks)
SFT	Ajuste Fino Supervisionado (Supervised Fine-Tuning)
SLM	Modelos de Linguagem Estatísticos (Statistical Language Models)
XAI	Inteligência Artificial Explicável (Explainable Artificial Intelligence)

SUMÁRIO

1 INTRODUÇÃO

O campo de pesquisa conhecido como Inteligência Artificial (IA), reconhecido como um dos domínios mais enigmáticos e promissores da ciência contemporânea, originado a partir dos avanços em Ciência de Dados e do Aprendizado de Máquina (AM), tem se consolidado como uma das áreas mais influentes dos últimos tempos. Seu impacto transcende a Tecnologia da Informação, promovendo transformações em campos como a medicina, a educação e, notoriamente, o direito — algo que será explorado ao longo deste trabalho.

A magnitude de seu impacto é tamanha que figuras proeminentes da indústria tecnológica vêm se pronunciando de forma audaciosa sobre suas expectativas em relação a essa tecnologia emergente. Sundar Pichai, CEO da Google, declarou que "a inteligência artificial é mais profunda do que a eletricidade ou fogo", enfatizando o potencial disruptivo dessa tecnologia IA.

Após períodos de estagnação, denominados "Inverno da Inteligencia Artificial" (RUSSELL; NORVIG, 2016) — marcados pela frustração de expectativa e pela retração de investimentos — o campo tem experimentado, nas últimas décadas, um crescimento exponencial. Esse avanço tem se destacado especialmente no subcampo conhecido como Processamento de Linguagem Natural (PLN), notoriamente com o advento dos Modelos de Linguagem de Grande Escala (LLMs)s, revolucionando a forma como sistemas computacionais interagem com a linguagem humana. Entre os modelos mais notáveis estão os da família GPT (Generative Pre-trained Transformer) e o LLaMA (Large Language Model Meta AI), ambos classificados como Modelos de Linguagem de Grande Escala (LLMs)s de propósito geral.

A eficiência dos LLMs na manipulação da linguagem natural deve-se, em grande parte, à arquitetura Transformers, proposta por Vaswani *et al.* (2017). Essa arquitetura é baseada no mecanismo de atenção, mais especificamente na chamada self-attention, que possibilita a análise contextual de cada elemento textual (token), independente da sua posição na sequência. Assim, cada token é processado não apenas considerando seu significado isolado, mas também seu contexto em relação aos outros tokens de entrada.

Esse modelo de processamento é crucial para tarefas de transdução de sequência, ou sequence-to-sequence, nas quais uma sequência de entrada é convertida em uma sequência de saída. Trata-se de uma estrutura essencial para aplicações avançadas no campo do PLN, como tradução automática, reconhecimento de fala, sumarização de textos, análise de sentimentos, geração de texto e reconhecimento de entidades nomeadas. A versatilidade dessa abordagem

tem possibilitado avanços significativos nos sistemas de automação, permitindo que tarefas historicamente atribuídas à cognição humana possam ser assumidas por sistemas computacionais.

Essa capacidade já vem sendo explorada em diversas áreas que lidam com uma vasta quantidade de dados e informações, muitas vezes na forma de documentos contendo texto não estruturado. No âmbito jurídico, ferramentas como DraftWise têm desenvolvido soluções baseadas em IA para auxiliar advogados na redação e negociação de contratos.

Além disso, estudos como o de Mahoney *et al.* (2019) propõem frameworks para classificação de texto aplicável no contexto da revisão de documentos legais, visando melhorar a eficiência e a transparência na identificação de documentos relevantes durante processos legais.

Notoriamente, o processo de documentação e confecção de documentos a partir de texto não estruturado, uma tarefa extremamente relevante para a área do direito, se alinha muito bem com as capacidades dos modelos de LLM e suas afinidade com tarefas de transdução de sequências. Escritórios e departamentos jurídicos elaboram diariamente documentos como petições, procurações, contratos, acordos e atas de reunião, uma tarefa de natureza complexa, agravada pelo alto fluxo de trabalho que qualifica a prática jurídica.

Portanto, diante desse contexto, a automação da redação de documentos jurídicos por meio de técnicas de NLP associadas a Modelos de Linguagem de Grande Escala mostra-se como uma solução promissora para solução programática para o desafio descrito. Essa abordagem visa não apenas otimizar o desempenho das atividades jurídicas, mas também promover eficiência, padronização e redução de erros em tarefas que demandam processamento intensivo de linguagem natural.

1.1 MOTIVAÇÃO

No contexto político e judiciário, a transparência e a audibilidade configuram-se como pilares essenciais para garantir a confiabilidade das informações e a continuidade na gestão dos processos institucionais (FERNANDES, 2021) não apenas sustentam a integridade das ações administrativas e jurídicas, como também promovem a devida prestação de contas à sociedade, elemento central em uma democracia.

Nesse cenário, as atas de reunião assumem um papel estratégico como instrumento formal de registro. Sua importância reside na capacidade de sintetizar, de forma clara, precisa e organizada, os acontecimentos relevantes de um encontro institucional, incluindo os temas debatidos, as decisões tomadas e os encaminhamentos definidos. Diferentemente de uma sim-

ples transcrição literal das falas, a ata oferece uma representação estruturada dos eventos, o que facilita sua posterior consulta e contribui diretamente para a governança e a memória organizacional.

Além da sua função administrativa e documental, as atas também desempenham um papel crucial no cumprimento das normas legais de transparência e acesso à informação. Sua elaboração e disponibilização pública estão em conformidade com legislações como a Lei de Acesso à Informação (BRASIL, 2011) e a Lei da Transparência (BRASIL, 2009), reforçando o compromisso institucional com a responsabilidade e o controle social.

Apesar de sua relevância, a produção de atas ainda é, majoritariamente, uma tarefa manual, morosa e sujeita a inconsistências de estilo, conteúdo e qualidade. A ausência de padronização entre diferentes órgãos e setores acentua esses desafios, dificultando a verificação, a interoperabilidade e o uso eficiente desses documentos. Diante desse cenário, torna-se evidente a necessidade de soluções que automatizem e otimizem esse processo, promovendo ganhos de tempo, padronização e conformidade legal.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

A proposta envolve a criação de uma solução capaz de processar transcrições estruturadas e extrair automaticamente informações essenciais, como: lista de participantes e seus respectivos cargos ou funções, pautas deliberadas, unidades proponentes, pautas extraordinárias sugeridas durante a reunião, assuntos discutidos e deliberações finais.

1.2.2 Objetivos Específicos

Com essa abordagem, busca-se promover maior agilidade, precisão e padronização na elaboração de atas, contribuindo para a transparência e a eficiência dos processos institucionais. Essa iniciativa torna-se especialmente relevante em contextos nos quais a documentação tem papel crítico, como nos ambientes jurídico e administrativo. Para alcançar o objetivo geral, o trabalho será orientado por três objetivos específicos:

a) Investigar o uso de modelos avançados de inteligência artificial na automação da extração de informações a partir de transcrições de reuniões
b) Analisar o impacto de diferentes configurações dos modelos e de prompts sobre

a qualidade das respostas geradas

c) Implementação do código fonte do sistema capaz de gerar atas a partir da transcrição de reuniões.

d) Testar e comparar o desempenho de diversos modelos de linguagem natural em condições controladas, avaliando critérios como precisão, consistência e adequação às necessidades do domínio jurídico.

Ao final da pesquisa, espera-se atingir dois resultados principais: (i) gera conclusões quanto ao desempenho dos modelos de LLM utilizados na tarefa de automação de Atas de reunião e como pode se aplicar para outros documentos; e (ii) a entrega de uma aplicação funcional, configurada como um produto mínimo viável (MVP), que integre todos os componentes desenvolvidos e comprove a viabilidade técnica e prática da solução. A aplicação deverá apresentar uma arquitetura escalável, eficiente e com potencial de uso real em ambientes institucionais.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção explora as principais teorias, conceitos e tecnologias relevantes para o desenvolvimento e a aplicação de técnicas de Inteligência Artificial (IA) e engenharia de prompt (prompting) no contexto da automação de documentos jurídicos.

Para compreender de forma mais aprofundada os modelos utilizados, os comportamentos observados e as estratégias de interação adotadas, torna-se essencial retomar os fundamentos teóricos e conceituais que sustentam os modelos de IA generativa. Inicialmente, serão apresentados os conceitos basilares de Inteligência Artificial (IA), Aprendizagem de Máquina (AM), Processamento de Linguagem Natural (PLN) e Engenharia de Prompt. Em seguida, direciona-se o foco para as técnicas e estruturas subjacentes aos modelos explorados neste trabalho.

Conceitos como Redes Neurais Artificiais (RNA), Aprendizagem Profunda (AP), mecanismos de atenção, especialmente a arquitetura Transformer, são fundamentais para a compreensão do funcionamento dos Modelos de Linguagem de Larga Escala (LLM) empregados no processo de automação aqui proposto. Esses modelos, amplamente utilizados em tarefas de geração textual, tradução automática e resumo de documentos, são capazes de captar relações contextuais complexas, mesmo em textos longos e com múltiplos interlocutores, como é o caso das transcrições de reuniões.

2.1 ATAS DE REUNIÃO

Atas são documentos que registram de forma clara, objetiva e precisa as ocorrências de uma reunião, assembleia ou convenção. Adquirem caráter oficial após sua aprovação formal, momento em que passam a ter validade jurídica (FERREIRA; CAMBRUSSI, 2015). A padronização na elaboração das atas é fundamental para assegurar sua função documental, sua clareza interpretativa e sua integridade jurídica.

Os componentes essenciais de uma ata incluem: cabeçalho, abertura, legalidade, referência aos presentes, aprovação da ata anterior, desenvolvimento e fecho. Cada um desses elementos possui função específica e contribui para a completude e validade do registro.

- **Cabeçalho:** Identifica a reunião, geralmente incluindo o título da ata, número sequencial e o tipo de evento (reunião ordinária, extraordinária, assembleia, sessão de julgamento etc.).
- **Abertura:** Indica o local, a data, o horário de início, bem como os responsáveis pela

condução e pela redação do documento.

- **Legalidade:** Refere-se à verificação e declaração do quórum mínimo exigido para a validade legal da reunião, conforme normas regimentais ou legais. A inexistência de quórum pode invalidar deliberações tomadas.
- **Referência aos Presentes:** Lista nominal dos participantes, geralmente acompanhada de seus respectivos cargos ou representações institucionais.
- **Aprovação da Ata Anterior:** Registra a leitura, discussão, correções e aprovação da ata da reunião anterior, quando aplicável (REPúBLICA, 2018).
- **Desenvolvimento:** Contém a narração objetiva e cronológica dos assuntos tratados, decisões tomadas, votações realizadas, posicionamentos apresentados e encaminhamentos definidos.

No contexto jurídico, a precisão e a completude desses elementos são fundamentais. A ata pode ser utilizada como prova documental em processos judiciais, servir de respaldo a recursos administrativos ou fundamentar decisões estratégicas, o que torna qualquer imprecisão um risco potencial à segurança jurídica. A Lei nž 8.159/1991, que institui a Política Nacional de Arquivos Públicos e Privados, estabelece a obrigação de preservação e correta gestão dos documentos arquivísticos, reconhecendo seu valor probatório e informacional (BRASIL, 1991).

Além disso, no cenário político e institucional, a transparência e a auditabilidade são princípios fundamentais para a legitimidade das ações públicas. Tais princípios são respaldados pela Lei nž 12.527/2011 (Lei de Acesso à Informação) (BRASIL, 2011) e pela Lei Complementar nž 131/2009 (Lei da Transparência) (BRASIL, 2009), que estabelecem diretrizes para a publicidade de documentos e atos administrativos.

As atas de reunião, portanto, constituem instrumentos formais de registro, com papel estratégico na governança institucional. Elas documentam, de maneira estruturada e objetiva, os debates, deliberações e decisões ocorridas, promovendo a memória organizacional e permitindo a prestação de contas à sociedade. Diferenciam-se de simples transcrições por sua capacidade de sintetizar informações de modo padronizado, facilitando a posterior recuperação e análise dos registros.

Contudo, apesar de sua relevância, a produção de atas ainda representa, na prática, uma tarefa majoritariamente manual, demandando tempo e sujeita a inconsistências de estilo, linguagem ou completude. A ausência de padronização entre instituições distintas também compromete sua interoperabilidade e dificulta análises automatizadas ou comparativas.

Nesse contexto, o uso de Modelos de Linguagem de Grande Escala (LLMs) surge

como uma alternativa promissora para automatizar a elaboração de atas, garantindo maior padronização, celeridade e acurácia na geração desses documentos. As seções seguintes apresentam as tecnologias e os conceitos por trás dessa abordagem, bem como uma seção dedicada exclusivamente à aplicação prática desenvolvida neste trabalho.

2.2 INTELIGENCIA ARTIFICIAL

A Inteligência Artificial (IA) é um campo multidisciplinar que cruza diversas áreas do conhecimento, como matemática, filosofia, psicologia e ciência da computação, na busca por replicar capacidades cognitivas humanas por meio de modelos matemáticos, lógicos e computacionais. Esse esforço envolve tanto a tentativa de modelar o raciocínio humano quanto o desenvolvimento de sistemas capazes de tomar decisões autônomas a partir de informações do ambiente. Segundo (RUSSELL; NORVIG, 2016), IA é o ramo da ciência da computação dedicado ao estudo e desenvolvimento de agentes inteligentes, sistemas que percebem seu ambiente e tomam ações que maximizam suas chances de atingir determinados objetivos.

Embora seja uma área relativamente recente no contexto científico, suas bases remontam ao período pós-Segunda Guerra Mundial, com contribuições de pensadores como Turing (1950), que propôs questionamentos sobre a possibilidade de máquinas pensarem um marco conceitual importante que originou o chamado Teste de Turing, ainda hoje referenciado nas discussões sobre inteligência em máquinas. O termo Inteligência Artificial foi oficialmente usado em 1956, durante a Conferência de Dartmouth, evento considerado o marco inicial da IA como disciplina formal (MCCARTHY *et al.*, 1955).

Apesar dos avanços do campo, ainda não há consenso definitivo sobre o que, de fato, constitui a inteligência artificial. Na literatura, as definições costumam se organizar em dois grandes eixos: de um lado, abordagens baseadas nos processos de pensamento, que se preocupam com como ocorre o raciocínio, e de outro, abordagens focadas no comportamento observado, em que a inteligência é avaliada pelas ações, independentemente dos processos internos que levaram a elas. Além disso, cada eixo pode ser subdividido de acordo com o objetivo pretendido: se busca a emulação do comportamento humano ou se prioriza a racionalidade ideal, ou seja, agentes que tomam decisões ótimas, mesmo que não operem de forma semelhante ao raciocínio humano.

Reconhecendo essas divisões, Russell e Norvig (2016) expõem quatro abordagens clássicas para definir a IA:

1. Pensar como um ser humano — modelar processos cognitivos humanos;
2. Agir como um ser humano — comportamento semelhante ao humano (ex.: Teste de Turing);
3. Pensar racionalmente — seguir processos lógicos e inferências corretas;
4. Agir racionalmente — tomar ações ótimas com base nas percepções e objetivos.

Com isso, é possível compreender como as abordagens em IA giram em torno de dotar agentes não humanos de faculdades relacionadas ao raciocínio ou de capacidades que envolvem a tomada de decisões inteligentes. Decisões essas que não foram previstas pelo responsável da arquitetura do sistema, mas inferidas através de aprendizagem feita em cima de dados e experiência.

Esses agentes inteligentes, diferentemente dos sistemas computacionais tradicionais, são capazes de operar de forma robusta em ambientes caracterizados por mudanças rápidas, incertezas e imprevisibilidade (WEISS, 2001).

Diante desse cenário, pode-se afirmar que a IA tem como objetivo central o desenvolvimento de sistemas autônomos, dotados de capacidade de raciocínio que, quando inseridos em ambientes cujas características sejam compatíveis com os padrões previamente capturados e aprendidos, são capazes de desempenhar atividades e tomar decisões que, até então, eram consideradas inerentes à cognição humana. Tais decisões são contextualizadas com base nas informações extraídas do ambiente por meio de mecanismos de percepção e interpretação.

Com essa linha de raciocínio, a IA se aproxima da definição apresentada por Boden (2018), segundo a qual o objetivo da área é fazer com que máquinas realizem tarefas que requerem inteligência humana.

Nesse sentido, a tarefa de elaboração de atas de reunião se caracteriza como uma atividade que exige faculdades que se alinham com as capacidades descritas dos agentes inteligentes — raciocínio, percepção do ambiente, identificação de padrões, captura de contexto, autonomia e adaptabilidade — sustentando, assim, uma visão otimista na implementação de sistemas inteligentes para automação desse processo.

2.3 APRENDIZADO DE MÁQUINA

O principal objeto de estudo do campo de Inteligência Artificial são os agentes inteligentes, sistemas computacionais capazes de realizar ações de forma autônoma dentro do ambiente em que estão inseridos. Ademais, imbuídos de inteligência, esses agentes são carac-

terizados pela reatividade, pró-atividade e flexibilidade para lidar com situações que não foram previstas quando concebidos (WEISS, 2001).

Seus atributos, especialmente a autonomia, são fundamentados na experiência, componente fundamental da arquitetura desses sistemas. Assim, a aprendizagem, que consiste na capacidade de melhorar a partir de experiências passadas, é o que confere a esses agentes a adaptabilidade necessária para operarem de forma eficaz em contextos diversos e em constante transformação. Essa capacidade de adaptação é formalmente estudada no campo do Aprendizado de Máquina (Machine Learning), um dos principais subdomínios da Inteligência Artificial.

Um sistema de aprendizado pode ser definido como um programa de computador que toma decisões com base em experiências acumuladas, construídas a partir de soluções bem-sucedidas de problemas anteriores (REZENDE, 2003). Nesse sentido, as técnicas de aprendizado de máquina funcionam como um meio de inserir conhecimento nos agentes, capacitando-os a realizar inferências lógicas, identificar padrões e agir de forma inteligente diante de novos problemas.

Uma entidade está aprendendo se sua performance em futuras tarefas melhora após realizar observações sobre o ambiente que está inserido. Aprendizado pode se mostrar de diversas formas, desde formas triviais, como demonstrado ao anotar um número de telefone, ao profundo, como demonstrado por Albert Einstein, que inferiu uma nova teoria do universo (RUSSELL; NORVIG, 2016).

No estudo sobre aprendizagem, são identificados diferentes paradigmas e formas de aprender. Ainda assim, é possível definir um problema de aprendizado de forma generalizada com base em três componentes essenciais: a tarefa a ser executada, a experiência a partir da qual o sistema aprende e a medida de desempenho utilizada para avaliar o progresso. Essa formulação foi proposta por Tom Mitchell, que define: Diz-se que um programa de computador aprende a partir da experiência E, com respeito a alguma classe de tarefas T e medida de desempenho P, se seu desempenho em T, medido por P, melhora com a experiência E (MITCHELL, 2013).

A partir dessa formulação, surgem os principais paradigmas de aprendizado de máquina, classificados de acordo com a forma como o sistema interage com os dados de entrada. Os três mais representativos são:

1. **Aprendizado supervisionado:** o sistema é treinado com um conjunto de dados rotulados, onde cada exemplo é composto por uma entrada e sua respectiva saída desejada. O objetivo é construir um modelo capaz de generalizar o conhecimento para dados novos,

prevendo saídas para entradas inéditas.

2. **Aprendizado não supervisionado:** os dados não possuem rótulos. O sistema busca estruturas ou padrões ocultos nos dados, sendo comum a aplicação de técnicas como agrupamento (clustering) e redução de dimensionalidade.
3. **Aprendizado por reforço:** o agente aprende a tomar decisões por meio da interação com o ambiente, recebendo recompensas ou punições de acordo com as ações que realiza, buscando maximizar a recompensa acumulada ao longo do tempo (SUTTON; BARTO, 2018).

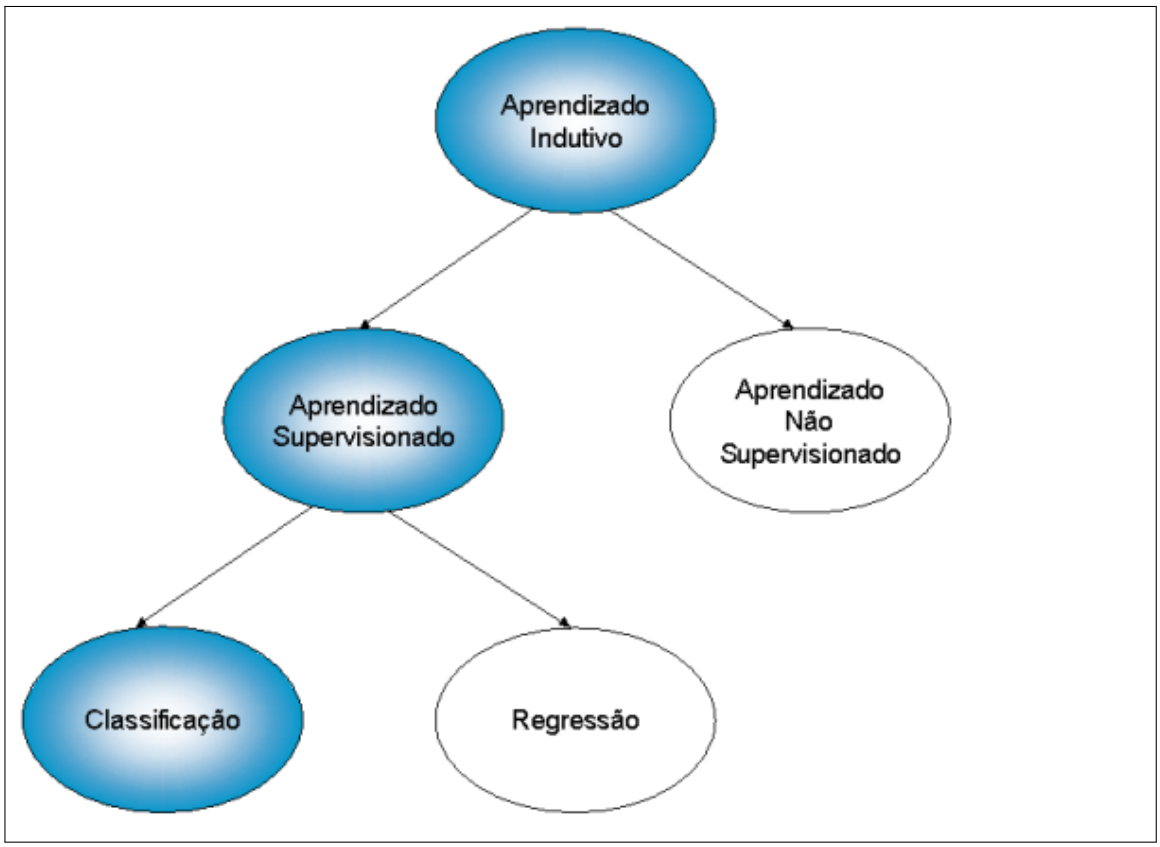
Cada um desses paradigmas se diferencia por diversos fatores, especialmente pelo tipo de feedback disponível no processo de aprendizagem, o que determina, por consequência, a forma como o conhecimento é adquirido pelo agente. No aprendizado supervisionado, por exemplo, o agente recebe uma coleção de exemplos, cada um composto por uma entrada observável e seu respectivo rótulo fornecido por um preceptor. Com base nesses dados e nos mecanismos de aprendizagem adotados, o agente aprende uma função que mapeia entradas para saídas de maneira consistente.

Esse tipo de problema, conhecido como aprendizado de entrada e saída (*input-output learning*), está entre os mais comuns e estudados dentro do campo da aprendizagem supervisionada. Nesse contexto, o objetivo central é construir uma função que relacione variáveis observáveis como imagens, textos ou vetores numéricos com saídas desejadas, que podem representar categorias, valores contínuos ou decisões. Segundo (RUSSELL; NORVIG, 2016), essa função é inferida a partir de exemplos rotulados e seu maior desafio está na generalização, ou seja, na capacidade do modelo de apresentar desempenho satisfatório em dados que não foram vistos durante o treinamento.

Dependendo da natureza dos rótulos de saída associados às entradas, os problemas supervisionados podem ser divididos em problemas de classificação ou de regressão, como pode ser observado na Figura 1 a seguir. Problemas de classificação ocorrem quando o objetivo do modelo é prever categorias discretas dentre um conjunto finito de valores, como acontece ao se tentar prever se um e-mail é spam ou não, ou identificar qual dígito foi escrito em uma imagem manuscrita. Quando há apenas duas categorias possíveis, como “sim” ou “não”, fala-se em classificação binária; já quando há múltiplas categorias, trata-se de classificação multiclasse (MITCHELL, 1997).

Por outro lado, em problemas de regressão, o valor a ser previsto é contínuo, pertencente a um domínio numérico não discreto. Casos típicos incluem a estimativa do valor de

Figura 1 – A hierarquia do Aprendizado



Fonte**:** (REZENDE, 2003)

imóveis com base em suas características ou a previsão de temperatura a partir de dados meteorológicos. Nesses casos, o modelo busca aproximar os valores reais por meio de uma função inferida dos dados (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

Além de sua aplicação em tarefas tradicionais de previsão, a regressão também serve de base conceitual para modelos generativos modernos, como os modelos de linguagem natural. Esses sistemas, fundamentados em arquiteturas de transformadores, são treinados para prever a próxima palavra (ou token) com base no contexto anterior, modelando distribuições condicionais sobre sequências de texto. Tarefas como language modeling (modelagem de linguagem) ou masked language modeling (modelagem de linguagem mascarada) são formuladas como problemas supervisionados em grandes corpora textuais, com pares entrada/saída derivados automaticamente (DEVLIN *et al.*, 2018).

A aprendizagem supervisionada pressupõe uma relação entre os valores de entrada e suas respectivas saídas. Do ponto de vista matemático, essa abordagem pode ser formalizada da seguinte maneira: dado um conjunto de treinamento com N pares de entrada-saída

$$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N),$$

onde cada y_j foi gerado por uma função desconhecida $f(x)$, o objetivo da aprendizagem é encontrar uma função h, denominada hipótese, que se aproxime da verdadeira função f. Assim, deseja-se que

$$h(x) \approx f(x)$$

para todos os exemplos, inclusive os ainda não vistos. Para medir a acurácia de uma hipótese, fornece-se a ela um conjunto de teste com exemplos distintos do conjunto de treinamento. Diz-se que uma hipótese generaliza bem se ela prediz corretamente o valor de y para exemplos novos.

Às vezes, a função f é estocástica ou seja, não é estritamente uma função de x e, nesse caso, o que se deseja aprender é uma distribuição de probabilidade condicional, $P(Y \mid x)$.

Em última instância, a aprendizagem supervisionada, como os demais paradigmas de aprendizado de máquina, contribui diretamente para o desenvolvimento de agentes inteligentes capazes de adaptar seu comportamento com base na experiência. Como sintetizado por Russell e Norvig (2016), o cerne da Inteligência Artificial está em projetar sistemas que possam agir racionalmente, isto é, tomar decisões que maximizem suas chances de sucesso com base em informações disponíveis. O aprendizado, nesse contexto, emerge como mecanismo fundamental para equipar esses agentes com a capacidade de inferir, generalizar e se aprimorar continuamente diante de novos desafios.

Portanto, ao entender a aprendizagem como um processo sistemático de aproximação entre dados e conhecimento, por meio da inferência de padrões e relações estatísticas, reafirma-se sua posição como um dos pilares da Inteligência Artificial moderna. Essa perspectiva amplia o alcance dos sistemas computacionais para além de tarefas programadas explicitamente, permitindo que eles operem de forma mais autônoma, robusta e eficaz em ambientes complexos e dinâmicos.

2.4 REDES NEURAIS ARTIFICIAIS (RNA)

Como foi visto na seção anterior, o campo de estudo do Aprendizado de Máquina (AM) tem como principal objeto o processo de aprendizagem em si, buscando não apenas com-

preender como ela ocorre, mas também desenvolver métodos que permitam incorporar essa capacidade aos sistemas computacionais. Boa parte dos esforços desse campo se concentra em introduzir nos algoritmos a habilidade de melhorar seu desempenho com base na experiência passada, de forma semelhante ao que ocorre com seres humanos ao longo do tempo.

Entre as abordagens mais influentes no contexto do AM, destacam-se as Redes Neurais Artificiais (RNAs), cujo desenvolvimento foi fortemente influenciado pelas descobertas no campo da neurociência. Graças a pesquisadores como Camillo Golgi e Santiago Ramón y Cajal, tornou-se possível estudar a estrutura e o funcionamento dos neurônios, as unidades básicas do sistema nervoso. Essas células, ao se interconectarem em redes complexas, demonstram ser capazes de realizar tarefas cognitivas fundamentais, além de estarem na base da formação da consciência (SEARLE, 1992).

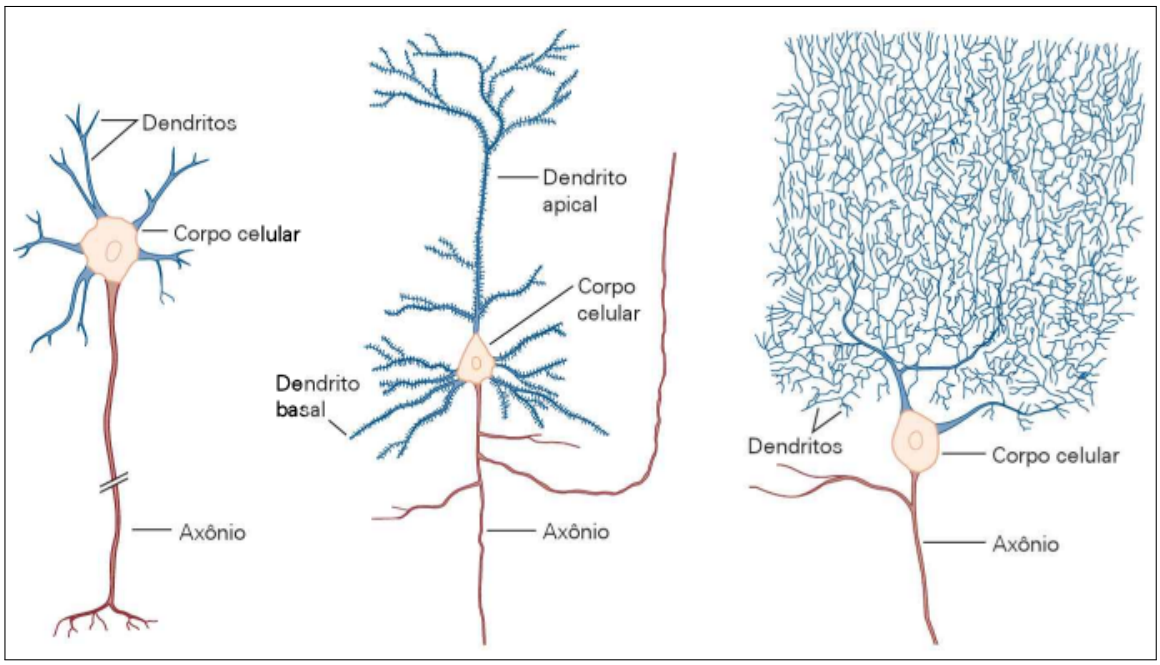
Na estrutura de uma rede neural biológica, os neurônios interagem entre si por meio de sinais, criando conexões chamadas sinapses. É através desse processo que estímulos externos são transformados em impulsos elétricos e químicos, que se propagam pela rede e ativam diferentes grupos de neurônios, cada um associado a funções específicas do sistema nervoso. Como resultado dessa ativação coordenada, o cérebro é capaz de gerar respostas complexas a partir de entradas sensoriais aparentemente simples (KANDEL, 2013).

A unidade básica de um neurônio é constituída pelo corpo da célula, ou soma, que contém o núcleo celular. Ramificando-se do corpo celular, encontramos numerosas fibras chamadas dendritos e uma única e longa fibra conhecida como axônio, como pode ser visto na Figura 2. O axônio pode se estender por uma distância considerável tipicamente 1 cm (100 vezes o diâmetro do corpo celular), mas pode chegar a até 1 metro de comprimento. Um neurônio pode fazer conexões com 10 a 100.000 outros neurônios nas junções sinápticas.

Os sinais são propagados de neurônio para neurônio por meio de uma complexa reação eletroquímica. Esses sinais controlam a atividade cerebral a curto prazo e também possibilitam mudanças de longo prazo na conectividade dos neurônios. Acredita-se que esses mecanismos formam a base do aprendizado no cérebro (RUSSELL; NORVIG, 2016).

Com o avanço no estudo do funcionamento do cérebro humano e a descrição detalhada das redes biológicas que o compõem, foi possível para pioneiros como Warren McCulloch e Walter Pitts propor, em 1943, um modelo matemático de neurônio artificial. Seu trabalho introduziu um sistema lógico baseado em redes de unidades binárias interconectadas, onde cada unidade imitava de forma abstrata o comportamento de um neurônio biológico, como se pode ver na Figura 3. Como sua contraparte biológica, o neurônio artificial emite um sinal de ativação

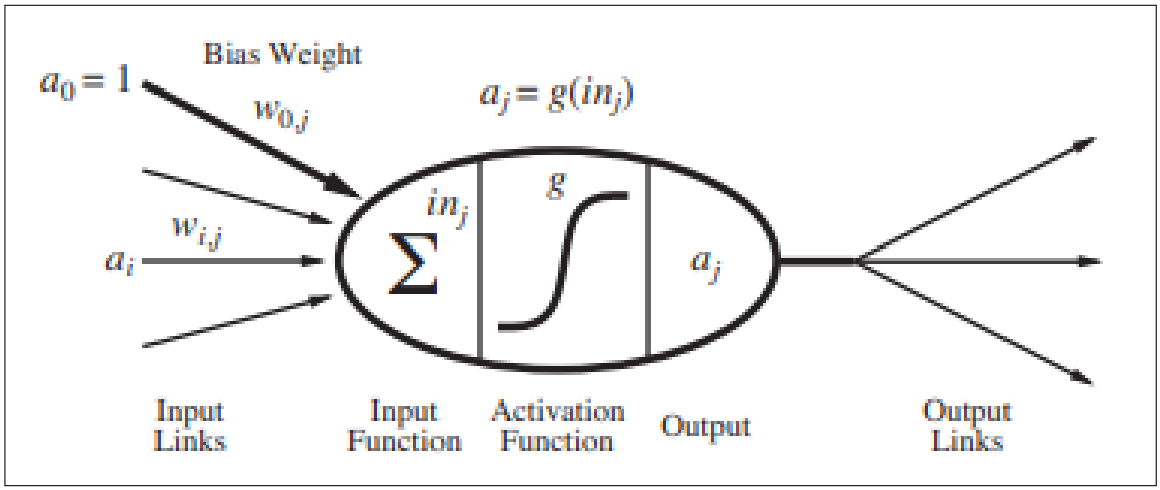
Figura 2 – Estrutura de um neurônio



Fonte**:** (KANDEL, 2013)

em resposta ao estímulo que recebe (MCCULLOCH; PITTS, 1943).

Figura 3 – Estrutura de um neurônio



Fonte**:** (RUSSELL; NORVIG, 2016)

O sinal é emitido quando o input recebido excede um limiar (threshold), que pode ser definido de forma rígida (hard threshold) ou flexível (soft threshold), dependendo do modelo adotado.

Esse comportamento é representado na modelagem através de uma função de ativação que recebe o resultado da combinação linear entre os pesos sinápticos e as respectivas componentes do input. Essa combinação pode ser expressa por:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

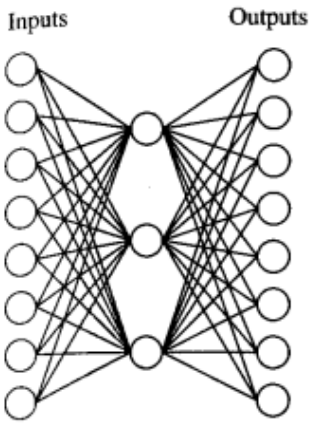
onde x_i representa as entradas do neurônio, w_i os pesos associados a cada entrada, b o termo de viés *bias* e f a função de ativação. Essa função tem o papel de introduzir não-linearidade no modelo, sendo essencial para que a rede neural artificial seja capaz de aprender relações complexas entre os dados de entrada.

Um dos modelos mais básicos de neurônio artificial é o perceptron, introduzido por (ROSENBLATT, 1958). O perceptron é caracterizado pelo uso de uma função de ativação do tipo limiar rígido (hard threshold), que emite uma saída binária dependendo se a soma ponderada das entradas ultrapassa ou não um determinado limiar. Dessa forma, a unidade produz um sinal ligado ou desligado, similar a um interruptor.

Partindo do perceptron isolado, as redes neurais são formadas pela conexão de múltiplas unidades, organizadas em camadas. Uma das diferentes formas de agrupar essas estruturas é a arquitetura MLP (Multilayer Perceptron), que possui conexões apenas unidirecionais — ou seja — forma um grafo direto e acíclico.

Nessa arquitetura, as unidades são organizadas em camadas distintas: a camada de entrada, uma ou mais camadas ocultas e a camada de saída (Figura 4). Cada neurônio de uma camada está conectado a neurônios da próxima camada, propagando a ativação de forma sequencial, sem ciclos de retorno.

Figura 4 – Rede Neural pós aprendizado



Input		Hidden Values				Output
10000000	→	.89	.04	.08	→	10000000
01000000	→	.15	.99	.99	→	01000000
00100000	→	.01	.97	.27	→	00100000
00010000	→	.99	.97	.71	→	00010000
00001000	→	.03	.05	.02	→	00001000
00000100	→	.01	.11	.88	→	00000100
00000010	→	.80	.01	.98	→	00000010
00000001	→	.60	.94	.01	→	00000001

Fonte**:** (MITCHELL, 2013)

Cada neurônio opera calculando uma combinação ponderada das ativações recebi-

das da camada anterior. Sobre essa soma ponderada, é aplicada uma função de ativação não linear, que introduz complexidade e permite à rede modelar relações não-lineares nos dados. O resultado dessa operação é então transmitido como entrada para os neurônios da próxima camada (RUSSELL; NORVIG, 2016). Essa estrutura hierárquica, composta por múltiplas camadas, possibilita que a rede construa representações progressivamente mais abstratas e complexas dos dados à medida que os sinais fluem através dela. Cada camada aprende a extrair características de um nível diferente de abstração, passando informações mais refinadas para a camada subsequente (GOODFELLOW; BENGIO; COURVILLE, 2016).

A ausência de ciclos simplifica o processo de treinamento e torna possível o uso do algoritmo de retropropagação para ajustar os pesos sinápticos. Este algoritmo é a espinha dorsal do aprendizado na maioria das redes neurais, pois permite ajustar os pesos sinápticos dos neurônios de forma a minimizar o erro entre a saída prevista pela rede e a saída desejada (o valor real ou o rótulo correto) (RUMELHART; HINTON; WILLIAMS, 1986).

O aprendizado efetivamente ocorre com o ajuste iterativo dos pesos em cada neurônio, em cada camada da rede. O processo começa com a propagação forward: os dados de entrada ativam a camada inicial e são processados sequencialmente através das camadas ocultas até atingirem a camada de saída, onde a rede gera uma predição. Em seguida, a etapa de retropropagação se inicia. O erro entre a predição da rede e o valor esperado é calculado e, a partir daí, é propagado no sentido inverso, da camada de saída de volta para as camadas de entrada. Durante essa propagação reversa, os pesos de cada conexão são atualizados de forma a reduzir gradualmente esse erro nas iterações futuras. É esse ciclo contínuo de propagação forward e retropropagação que permite à rede "aprender"a partir dos dados.

Assim, uma das principais características de uma rede neural é sua intrínseca capacidade de aprender, ou seja, sua habilidade de se aproximar e modelar funções complexas. A quantidade e a complexidade das funções que podem ser representadas por essa arquitetura dependem diretamente do número de camadas (profundidade da rede) e da quantidade de neurônios em cada uma delas, o que permite que redes neurais, especialmente as profundas, aprendam representações incrivelmente sofisticadas e não-lineares dos dados (MITCHELL, 2013).

2.5 APRENDIZADO PROFUNDO

De acordo com o Paradoxo de Morovac, articulado na obra Mind Child por Moravec (1995), tarefas que envolvem raciocínio lógico complexo, muitas vezes difíceis para seres

humanos, são relativamente fáceis para sistemas computacionais, enquanto for possível uma descrição clara da tarefa. Exemplos claros incluem o fato de que jogar xadrez em nível de grande mestre ou resolver equações matemáticas avançadas são comparativamente simples de programar para computadores. No entanto, localizar objetos em uma imagem, compreender a fala em um ambiente ruidoso e manipular objetos na vida real, tarefas realizadas instintivamente por humanos e difíceis de descrever através de parâmetros, mostraram-se extraordinariamente difíceis de replicar em máquinas.

É nesse contexto que redes neurais artificiais, como apresentado na sessão anterior, emergem como uma abordagem promissora. A hipótese central por trás das redes neurais é que qualquer problema complexo pode ser aproximado por uma função matemática. Dada uma quantidade suficientemente grande de dados de treinamento e uma arquitetura adequada, redes neurais com múltiplas camadas ocultas (as chamadas redes "profundas") demonstram uma notável capacidade de aprender e se aproximar de funções de ordem extremamente complexa. Esta propriedade fundamental é formalizada pelo Teorema da Aproximação Universal (HORNIK; STINCHCOMBE; WHITE, 1989).

A capacidade de representação dessas redes profundas provém justamente da presença de múltiplas camadas ocultas, que possibilitam a extração de padrões e abstrações em diferentes níveis de complexidade. No entanto, apesar do seu potencial teórico, durante muitos anos o treinamento eficaz dessas redes permaneceu um desafio, principalmente pela dificuldade em ajustar os pesos das camadas intermediárias de forma eficiente.

Somente a partir de 1986, com a publicação pelo autor Rumelhart, Hinton e Williams (1986) do artigo "Learning Internal Representations by Error Propagation", esse obstáculo pode ser superado. Nesse trabalho, foi apresentado o algoritmo de retropropagação do erro (backpropagation), que tornou viável o treinamento de redes com múltiplas camadas. A técnica baseia-se no uso do gradiente do erro para atualizar os pesos da rede, propagando esse gradiente da camada de saída para as camadas anteriores.

Esse algoritmo é dividido em dois processos complementares: a propagação direta (forward propagation) e a retropropagação do erro (backward propagation). Na primeira etapa, os dados de entrada percorrem a rede camada por camada, sendo transformados por combinações lineares seguidas da aplicação de funções de ativação não lineares, até a obtenção da saída final. Em seguida, calcula-se o erro, que corresponde à diferença entre a saída obtida e a saída esperada.

Na segunda etapa, esse erro é propagado de volta pelas camadas da rede, utilizando

o método do gradiente descendente para ajustar os pesos de forma a minimizar uma função de custo. Essa atualização ocorre iterativamente, permitindo que a rede aprenda a partir dos exemplos apresentados. Com o tempo, a rede se ajusta para realizar previsões mais precisas.

Apesar do avanço fundamental proporcionado pelo algoritmo de retropropagação, o potencial total das redes neurais profundas permaneceu, por muito tempo, inexplorado. Foi somente em 2012, com o sucesso da arquitetura AlexNet no desafio ImageNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2017), que o aprendizado profundo passou a ganhar notoriedade. A partir desse avanço, diversos fatores convergiram para alavancar o desenvolvimento dessa área.

De acordo com LeCun, Bengio e Hinton (2015), três fatores principais foram determinantes para essa ascensão:

- **Uso Eficiente de GPUs:** A capacidade de processar grandes volumes de dados em paralelo, utilizando unidades de processamento gráfico (GPUs), tornou o treinamento de modelos profundos computacionalmente viável e rápido.
- **Inovações Algorítmicas:** A adoção de funções de ativação mais eficientes, como as Retified Linear Units (ReLUs), que ajudaram a mitigar problemas de gradiente em redes profundas, e o desenvolvimento de novas técnicas de regularização, como o Dropout, que previne o overfitting, foram cruciais.
- **Aumento de Dados:** A capacidade de gerar mais exemplos de treinamento através de técnicas de aumento de dados (data augmentation), que deformam os dados existentes (como rotações, cortes e espelhamentos de imagens), expandiu significativamente os conjuntos de dados disponíveis e melhorou a generalização dos modelos.

A partir desses avanços combinados, a aprendizagem profunda não apenas superou as limitações históricas do aprendizado de máquina, mas também impulsionou um salto qualitativo na capacidade da IA de lidar com tarefas de percepção complexas, como visão computacional, reconhecimento de voz e recomendação de conteúdos.

Um dos campos mais impactados por essa evolução foi o Processamento de Linguagem Natural (PLN), que até então enfrentava grandes obstáculos na modelagem de estruturas linguísticas altamente variáveis e ambíguas. Inicialmente, o uso de Redes Neurais Recorrentes (RNNs) e suas variantes, como as LSTMs (Long Short-Term Memory) (HOCHREITER; SCHMIDHUBER, 1997), marcou um avanço importante na modelagem sequencial de texto, permitindo que padrões temporais e dependências de longo prazo fossem aprendidos diretamente a partir de grandes volumes de dados linguísticos. No entanto, mesmo com as melhorias, as RNNs e LSTMs possuíam limitações intrínsecas, principalmente relacionadas à dificuldade

de capturar dependências de longo prazo em sequências muito extensas e à dificuldade de paralelização de seu treinamento em larga escala.

Como veremos mais adiante, uma arquitetura inovadora surgiria para transformar o campo do PLN mais uma vez, superando essas barreiras e pavimentando o caminho para os modelos de linguagem que conhecemos hoje.

2.6 PROCESSAMENTO DE LINGUAGEM NATURAL

A linguagem, tanto escrita quanto falada, é um dos diversos atributos que distingue o Homo sapiens de todas as outras espécies que andam, voam, nadam ou rastejam na superfície da terra. Por volta de 100.000 anos atrás, humanos aprenderam a falar, e por volta de 7.000 anos atrás, a escrever. Apesar de haver espécies capazes de demonstrar vocabulário de centenas de sinais, apenas humanos foram capazes de desenvolver e usar linguagens para comunicar um número indeterminado de mensagens e ideias (RUSSELL; NORVIG, 2016).

Linguagens formais, como as de programação (Java e Python), possuem regras e uma semântica rígida que delimitam um número infinito de programas válidos, chamados de gramática. Essas linguagens são incorporadas nos sistemas computacionais e podem ter seu sentido interpretado e traduzido diretamente para a linguagem nativa das máquinas. No entanto, as linguagens naturais, como Português e Inglês, são intrinsecamente mais complexas e ambíguas. Elas não seguem um conjunto de regras fixas e facilmente formalizáveis; em vez disso, evoluem organicamente e dependem fortemente do contexto, da cultura, das nuances de entonação e da intenção do falante para que seu significado seja plenamente compreendido (JURAFSKY; MARTIN, 2009). A mesma palavra pode ter múltiplos significados, e a estrutura de uma frase pode variar enormemente sem alterar seu sentido essencial, tornando a interpretação computacional um desafio substancial.

Assim, para que a tarefa de análise de linguagem natural seja feita por máquinas, além de compreender os símbolos que compõem a língua, é preciso que seu sentido seja descrito de forma clara e sem ambiguidade, através de regras e padrões, para o sistema computacional, de modo que seu significado possa ser devidamente capturado algo próximo do que é feito com as linguagens formais. Nesse sentido, podemos entender o PLN como uma área interdisciplinar que reúne conhecimentos da Ciência da Computação, Inteligência Artificial e Linguística. Seu objetivo é desenvolver métodos e algoritmos que permitam às máquinas não apenas processar o texto ou a fala, mas também compreender as intenções e o significado por trás da comunicação

humana

Historicamente, as primeiras tentativas de fazer computadores entenderem a linguagem foram fundamentadas em regras gramaticais explícitas, elaboradas programaticamente e inseridas a mão nos sistemas responsáveis pelo processamento. No entanto, a natureza ambígua e irrestrita inerente à linguagem natural, além das inflexões que toda linguagem sofre quando usadas de fato, revelaram a impraticabilidade dessa abordagem puramente simbólica e artesanal (NADKARNI; OHNO-MACHADO; CHAPMAN, 2011).

Processamento de Linguagem natural deve extrair o sentido (semântica) do texto, assim, um dos principais desafios enfrentados era: As gramáticas formais especificam principalmente a sintaxe (a relação entre unidades de texto, como substantivos, verbos e adjetivos). Embora fosse possível estender essas gramáticas para abordar a semântica da linguagem natural através da expansão massiva de subcategorias e regras adicionais (por exemplo, a regra de que o verbo "comer"se aplica apenas a substantivos de itens ingeríveis) (NADKARNI; OHNO-MACHADO; CHAPMAN, 2011), isso levava a dois problemas críticos:

1. **Explosão de Regras e Interações Imprevisíveis:** As regras tornavam-se incontrolavelmente numerosas e, muitas vezes, interagiam de maneira imprevisível, levando a um aumento na frequência de "interpretações ambíguas"(múltiplas interpretações possíveis para uma sequência de palavras). Exemplos como trocadilhos (ambiguidades usadas para efeito humorístico) demonstram a sutileza com que humanos lidam com a polissemia, algo que sistemas baseados em regras falhavam.
2. **Inflexibilidade com Variações Linguísticas:** Regras escritas à mão lidavam muito mal com a prosa não-gramatical da fala cotidiana e com textos altamente telegráficos (como notas de progresso hospitalares em contextos médicos), que, embora compreendidos por humanos, desviavam-se das estruturas formais esperadas.

Essas limitações impulsionaram o surgimento do PLN Estatístico no final do século XX. Essa nova abordagem focava em aprender padrões a partir de grandes corpora de texto, utilizando modelos probabilísticos e técnicas de aprendizado de máquina. Em vez de regras explícitas, os sistemas estatísticos inferiam as probabilidades de ocorrência de palavras e sequências, permitindo uma maior robustez à variabilidade e ambiguidade da linguagem real.

Uma das grandes sacadas das últimas cinco décadas de pesquisa em processamento de linguagem é que o vasto e complexo conhecimento linguístico pode ser capturado por um número limitado de modelos formais e teorias. Felizmente, esses modelos são derivados de ferramentas padrão da ciência da computação, matemática e linguística (JURAFSKY; MARTIN,

2009). Entre os mais importantes estão:

- **Máquinas de Estado:** Em sua formulação mais simples, são modelos formais que consistem em estados, transições entre estados e uma representação de entrada. Variações comuns incluem autômatos finitos determinísticos e não determinísticos, e transdutores de estados finitos. Esses modelos são cruciais para lidar com o conhecimento de fonologia, morfologia e sintaxe, frequentemente complementados por sistemas de regras formais, como gramáticas regulares e livres de contexto.
- **Lógica:** Modelos baseados em lógica, como a lógica de primeira ordem (cálculo de predicados), têm sido tradicionalmente empregados para modelar a semântica e a pragmática da linguagem. Eles visam representar o significado e as relações lógicas dentro de sentenças, embora abordagens mais recentes tenham se voltado para técnicas mais robustas da semântica lexical não-lógica.
- **Modelos Probabilísticos:** Cruciais para capturar todo tipo de conhecimento linguístico, esses modelos permitem que cada uma das abordagens anteriores (máquinas de estado, sistemas de regras e lógica) seja aumentada com probabilidades. Por exemplo, uma máquina de estado pode se tornar um autômato ponderado ou um Modelo de Markov. Dentre eles, os Modelos de Markov Ocultos (*HMMs - Hidden Markov Models*) foram amplamente utilizados em diversas aplicações do PLN, como marcação de classes gramaticais *(part-of-speech tagging*), reconhecimento de fala, compreensão de diálogo e tradução automática. A principal vantagem dos modelos probabilísticos é sua capacidade de resolver os diversos problemas de ambiguidade; quase qualquer problema de processamento de fala e linguagem pode ser reformulado como: "dadas N escolhas para alguma entrada ambígua, escolha a mais provável"(JURAFSKY; MARTIN, 2009).
- **Modelos de Espaço Vetorial:** Baseados em álgebra linear, esses modelos são a base da recuperação de informação e de muitos tratamentos do significado das palavras, onde palavras e documentos são representados como vetores numéricos em um espaço multidimensional. A proximidade entre vetores indica similaridade semântica.

No entanto, os avanços recentes no campo do aprendizado profundo provocaram uma verdadeira revolução na forma como os sistemas computacionais compreendem e geram linguagem natural. Enquanto os modelos estatísticos clássicos dependiam significativamente de engenharia de características ou seja, da extração manual de atributos linguísticos relevantes para alimentar os modelos , as redes neurais profundas trouxeram a capacidade de aprender automaticamente essas representações diretamente a partir dos dados brutos (LECUN; BENGIO;

HINTON, 2015; GOODFELLOW; BENGIO; COURVILLE, 2016). Essa abordagem tornou possível capturar padrões complexos e hierarquias de significado que antes eram inacessíveis ou exigiam considerável intervenção humana.

Dentre os principais marcos dessa transformação, destacam-se os modelos baseados em Redes Neurais Recorrentes (RNNs), especialmente suas variantes mais sofisticadas, como as LSTMs (Long Short-Term Memory) (HOCHREITER; SCHMIDHUBER, 1997) e as GRUs (Gated Recurrent Units). Essas arquiteturas foram desenvolvidas para lidar com dados sequenciais e se mostraram eficazes ao modelar dependências temporais e contextuais, superando muitas limitações dos modelos probabilísticos tradicionais. Elas obtiveram destaque em diversas tarefas de Processamento de Linguagem Natural, como tradução automática, sumarização, classificação de sentimentos e resposta a perguntas.

Apesar dos avanços, as RNRs ainda apresentavam limitações estruturais importantes. Por dependerem de um processamento estritamente sequencial palavra por palavra , essas redes impunham restrições significativas à paralelização durante o treinamento, além de apresentarem dificuldades em capturar relações de longa distância dentro dos textos. Esses desafios se tornaram ainda mais críticos com o aumento exponencial do volume de dados e da complexidade das tarefas linguísticas modernas.

Foi nesse contexto de crescente demanda por modelos mais eficientes e expressivos que emergiu uma nova classe de arquiteturas que dispensam a recorrência e introduzem mecanismos mais flexíveis de processamento sequencial. Nas próximas seções será exposto em detetalhes como esse no paradigma redefiniu o estado da arte no PLN e o que faz dele superior quando comparado ao seu predecessor, as Redes Neurais Recorrentes.

2.7 REDES NEURAIS RECORRENTES

Para entender melhor o impacto execido pela arquitetura que deu origem aos modelos de linguagem modernos, é preciso entender as capacidades de limitações de seu predecessor.

As Redes Neurais Recorrentes (*Recurrent Neural Networks, RNNs*) representam uma classe de arquiteturas desenvolvidas para o tratamento de dados sequenciais, como texto, fala ou séries temporais numéricas, onde a ordem e a dependência temporal entre os elementos importam. Ao contrário das redes *Feedforward* (como as MLPs), que processam dados seguindo um fluxo linear sem ciclos, as RNNs incorporam um laço de feedback (conexões recorrentes) que permitem a persistência da informação ao longo do tempo (SCHMIDT, 2019).

Esse comportamento característico das RNNs é determinado pela estrutura chaamda Célula Recorrente, que opera como um vetor de estado interno (*s[n]*), agindo como uma memória de curto prazo. Sua definição, baseada a partir de sistemas dinâmicos contínuos, se da atravês da seguinte equação canônica:

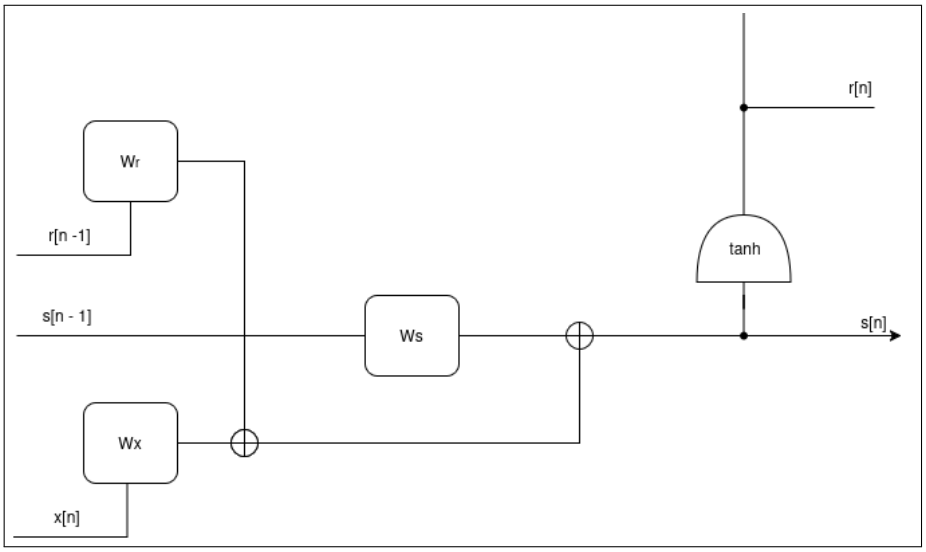
$$\vec{s}[n] = W_s\vec{s}[n-1] + W_r\vec{r}[n-1] + W_x\vec{x}[n] + \vec{\theta}_s$$

Onde:

- $\vec{s}[n]$: É o novo estado interno (a nova memória)
- $\vec{s}[n-1]er[\vec{n-1}]$: A memória anterior e seu sinal de leitura (readout) (estado da rede no passo de tempo n1).
- $\vec{x}[n]$: A entrada atual (o embedding da palavra no passo n).
- W_s, W_r e W_x: Matrizes de pesos compartilhados que ponderam a contribuição da memória anterior, do readout e da entrada atual.
- $\vec{\theta}_s$: Vetor de bias.

O processo de cálculo é estritamente sequencial. Para cada passo de tempo n, a rede precisa da entrada $x[n]$ e, crucialmente, do estado interno imediatamente anterior $s[n-1]$. Para fins de treinamento, esse laço recorrente é "desdobrado"(*unrolled*) ao longo de toda a sequência, transformando conceitualmente em uma rede *feedforward* profunda ao longo do tempo, onde cada passo de tempo é uma nova camada que reutiliza os mesmos pesos (SHERSTINSKY, 2023).

Figura 5 – Diagrama de Fluxo da Célula Canônica da Recurrent Neural Network (RNN).



Fonte**:** (SHERSTINSKY, 2023)

O Diagrama de Fluxo da Célula Canônica (Figura 5) ilustra esse mecanismo de processamento, onde o novo estado $\vec{s}[n]$ é computado como uma combinação linear de três componentes ponderados: a entrada atual $\vec{x}[n]$ ($W_x\vec{x}[n]$), o readout anterior $\vec{r}[n-1]$ ($W_r\vec{r}[n-1]$) e o estado anterior $\vec{s}[n-1]$ ($W_s\vec{s}[n-1]$). Essa soma linear é então submetida a uma função de ativação não-linear (G, como a tanh) para gerar o sinal de leitura $\vec{r}[n]$, que será reutilizado no próximo passo de tempo, perpetuando assim a memória do sistema.

Assim como na rede *Multilayer Perceptron*, o treinamento desse modelo usa uma variação do algortimo *backpropagation*, denominada *Back Propagation Through Time* (BPTT), com o objetivo de propagar o erro e calcular os gradientes para atualizar os pesos (SCHMIDT, 2019).

O BPTT trata a RNN desdobrada por K passos como uma rede *feedforward* profunda com K camadas, onde o conjunto de pesos (W_x, W_r, W_s) é compartilhado em todas elas. O gradiente total para cada parâmetro é, portanto, a soma dos gradientes calculados em cada passo de tempo da sequência. Essa soma é calculada na fase de propagação para trás (*backward pass*), onde o erro é retro-propagado recursivamente a partir do último passo da sequência.

No entanto, apesar de ser teoricamente possível o treinamento de uma RNN, o BPTT, quando aplicado a sequências longas, apresenta uma instabilidade numérica causada pela sua natureza de laço de repetição, que impede o modelo de aprender dependências distantes.

Esse problema está liagado à à regra da cadeia multiplicativa que rege a propação a propagação do gradiente do erro ($\vec{\psi}$) através de muitos passos de tempo, conforme dado pela a seguir:

$$\frac{\partial \vec{\psi}[n]}{\partial \vec{\psi}[l]} = \prod_{k=n+1}^{l} W_r \odot \frac{dG(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{s}[k]}$$

Característico das RNNs, esse defeito é conhecido como *Exploding \Vanishing Gradient Problem* (VENNERøD; KJæRRAN; BUGGE, 2021). É dito que, em teoria, o estado oculto de cada célula RNN pode rastrear a informação por um tmepo arbitrário. No entanto, conforme o valor oculto de $s[n]$ é atualizado a cada passo de n, a depedência do tempo acarreta nas depdências de longo prazo (para sequências acima de 100 passos) serem ignoradas durante o treino (LE; ZUIDEMA, 2016).

O termo $\prod_{k=n+1}^{l} W_r \odot \frac{dG(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{s}[k]}$ na equação acima demonstra que a influência do gradiente em um passo distante (l) sobre um passo anterior (n) é determinada por uma

multiplicação em cadeia de matrizes Jacobianas. Essa multiplicação resulta em dois problemas críticos:

1. Desaparecimento de Gradientes (*Vanishing Gradients*): Ocorre quando os valores dominantes (autovalores) da matriz de peso recorrente W_r e a derivada da função de ativação G são consistentemente menores que 1. Conforme o gradiente é retro-propagado, o produto em cadeia tende a zero exponencialmente, fazendo com que os pesos nos primeiros passos da sequência não recebam atualizações significativas.
2. Explosão de Gradientes (*Exploding Gradients*): Ocorre se os autovalores da matriz W_r forem consistentemente maiores que 1. Neste cenário, a magnitude do gradiente cresce exponencialmente, o que leva à instabilidade no treinamento e a grandes *overflows* numéricos. A mitigação prática para este problema é a técnica de *gradient clipping*, que limita o valor máximo do gradiente.

O problema do *vanishing gradient* foi o que inspirou o desenvolvimento da arquitetura *Long Short-Term Memory* (LSTM) em 1997 (HOCHREITER; SCHMIDHUBER, 1997). A LSTM abordou essa falha estrutural ao introduzir o caminho de estado de célula C_t (ou $s[n]$ em uma notação simplificada), que permite que o gradiente flua de forma estável através de operações de soma em vez de multiplicação, criando o modo "*Constant Error Carousel* (CEC)". Adicionalmente, a LSTM utiliza portões multiplicativos (*gates*) (como o *Forget* e *Input* gate) que controlam seletivamente quais informações reter ou descartar, garantindo que a rede possa aprender e reter dependências de longo prazo.

Apesar de as LSTMs representarem um avanço significativo, permitindo o aprendizado de dependências mais longas, elas ainda enfrentam restrições estruturais e computacionais. O processamento estritamente sequencial das RNNs e LSTMs implica que cada passo de tempo depende do anterior, impossibilitando a paralelização do cálculo durante o treinamento. Isso resulta em altos custos computacionais e baixa escalabilidade quando aplicadas a grandes volumes de dados, como aqueles necessários para modelagem de linguagem em larga escala (VASWANI *et al.*, 2017).

Além disso, mesmo com os mecanismos de portões, o alcance das dependências de longo prazo ainda é limitado especialmente em sequências muito extensas, nas quais a informação relevante precisa ser propagada por centenas ou milhares de passos. Esse comportamento se reflete em dificuldades de captura de relações contextuais distantes, como, por exemplo, dependências gramaticais entre palavras separadas por grandes trechos de texto.

Em resposta a essas limitações, surgiram variações como as Gated Recurrent Units

(GRU) (CHO *et al.*, 2014), que simplificam a estrutura da LSTM mantendo desempenho similar, e arquiteturas híbridas com mecanismos de atenção adicionados sobre camadas recorrentes (*Attention-based RNNs*). No entanto, a presença de recorrência ainda impõe uma barreira intrínseca à eficiência e à capacidade de paralelização.

Essa busca por maior eficiência e por uma modelagem mais direta das relações entre todos os elementos de uma sequência levou, em 2017, à introdução da arquitetura Transformer (VASWANI *et al.*, 2017). O Transformer elimina completamente a recorrência, substituindo-a por um mecanismo de atenção auto-regressiva (*self-attention*) capaz de modelar dependências globais entre quaisquer posições da sequência de forma paralela. Essa abordagem não apenas resolve os problemas de gradiente e paralelização, mas também se mostrou excepcionalmente escalável, constituindo a base dos Modelos de Linguagem de Grande Escala (LLMs) modernos.

2.8 ARQUITETURA TRANSFORMER

Como discutimos na seção anterior, o campo de Processamento de Linguagem Natural (PLN) teve avanços significativos com o surgimento do aprendizado profundo. Tarefas clássicas da área, como análise de sentimentos, reconhecimento de entidades nomeadas e tradução automática, passaram a ser tratadas com redes neurais, especialmente as recorrentes (ANSAR; GOSWAMI; CHAKRABARTI, 2024). Contudo, essas arquiteturas enfrentavam desafios consideráveis, como a dificuldade de capturar dependências de longo prazo em sequências muito extensas e a ineficiência no treinamento em larga escala devido à sua natureza sequencial. O verdadeiro divisor de águas no desempenho desses sistemas, que superou essas limitações e redefiniu o campo, foi a introdução da arquitetura conhecida como *Transformer*.

Proposta em 2017 por Vaswani e colaboradores, no artigo seminal *Attention Is All You Need* (VASWANI *et al.*, 2017), a arquitetura Transformer revolucionou a forma como os modelos processam sequências de linguagem. Ela se tornou a base para o desenvolvimento de modelos amplamente utilizados atualmente, como o *Bidirectional Encoder Representations from Transformers* (BERT) e os *Generative Pre-trained Transformers* (GPT) (TOPAL; BAS; HEERDEN, 2021).

O principal diferencial do *Transformer* está no uso intensivo do mecanismo de atenção, que permite ao modelo avaliar a importância relativa de cada palavra no contexto de uma sentença. Isso possibilita a codificação de relações complexas entre palavras, mesmo quando estão distantes entre si na sequência textual. Ao contrário das redes recorrentes, que processam

a entrada de forma sequencial, essa nova arquitetura opera de forma paralela, o que o torna substancialmente mais eficiente para tarefas de treinamento em larga escala (VASWANI *et al.*, 2017).

Para compreender plenamente o impacto dessa abordagem e como o *Transformer* avalia a significância das palavras, é necessário primeiro entender o processo fundamental de codificação de palavras e como essa representação evoluiu das arquiteturas anteriores. Durante os anos 2000, como uma maneira de suprir a incapacidade dos modelos de linguagem estatísticos tradicionais em lidar com a relação semântica entre conceitos e o contexto da linguagem, foi desenvolvida a técnica de *embedding* (ANSAR; GOSWAMI; CHAKRABARTI, 2024).

A técnica de *word embedding* consiste em representar uma palavra num espaço vetorial contínuo de alta dimensionalidade. Nesses espaços, palavras com significados semelhantes ou que ocorrem em contextos parecidos são mapeadas para vetores numericamente próximos. Essa proximidade vetorial captura relações semânticas e sintáticas, permitindo que os modelos de PLN interpretem o "sentido"das palavras de uma forma que as abordagens simbólicas ou *one-hot* não conseguiam.

Modelos como Word2Vec (MIKOLOV *et al.*, 2013) e GloVe (PENNINGTON; SOCHER; MANNING, 2014) tornaram-se fundamentais nesse processo, permitindo a captura de relações léxicas e semânticas de maneira eficiente. Ao invés de tratar cada palavra como uma entidade discreta, como nos métodos de representação *one-hot* (onde cada palavra no vocabulário recebia um vetor binário com um "1"na posição correspondente e "0"nas demais, resultando em vetores esparsos sem relações semânticas intrínsecas), os *embeddings* incorporaram informação contextual distribuída, extraída de grandes corpora de texto. Essa mudança permitiu que os modelos compreendessem, por exemplo, que "rei"e "rainha"estão relacionados por um padrão vetorial semelhante ao de "homem"e "mulher", tornando as representações mais ricas e significativas para o processamento computacional da linguagem.

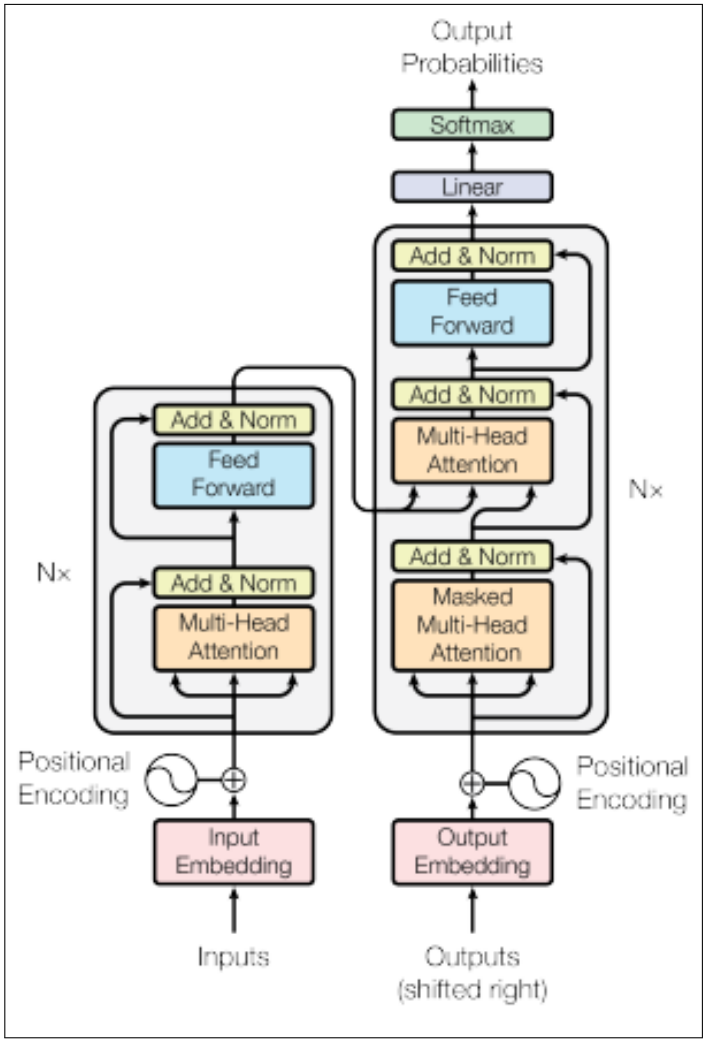
A representação numérica de palavras por meio de embeddings não apenas ampliou a capacidade das máquinas de capturar nuances semânticas, como também reduziu significativamente o custo computacional, ao projetar essas palavras em um espaço vetorial mais compacto e denso. Modelos especializados na geração desses embeddings passaram a ser amplamente integrados a arquiteturas do tipo Encoder-Decoder, especialmente aquelas baseadas em Redes Neurais Recorrentes (RNNs) e em suas variantes mais sofisticadas, como as LSTMs e GRUs. Nesses modelos, o encoder processa a sequência de entrada palavra por palavra, transformando os embeddings em um vetor de contexto que resume as informações relevantes da sentença (FU

et al., 2023).

A estrutura Encoder-Decoder constitui a base da maioria dos modelos neurais de transdução de sequência mais competitivos (VASWANI *et al.*, 2017). Nessa arquitetura, o encoder recebe como entrada uma sequência de representações simbólicas $(x_1, x_2, \ldots, x_n)$ e a transforma em uma sequência de representações contínuas $\mathbf{z} = (z_1, z_2, \ldots, z_n)$, as quais servem como base para a etapa de decodificação realizada pelo decoder.

A arquitetura Transformer adota essa estrutura geral, substituindo a recorrência por mecanismos de autoatenção empilhados e camadas totalmente conectadas (fully connected, aplicadas ponto a ponto), tanto no encoder quanto no decoder, como ilustrado nas metades esquerda e direita da Figura 6, respectivamente.

Figura 6 – Arquitetura Transformers



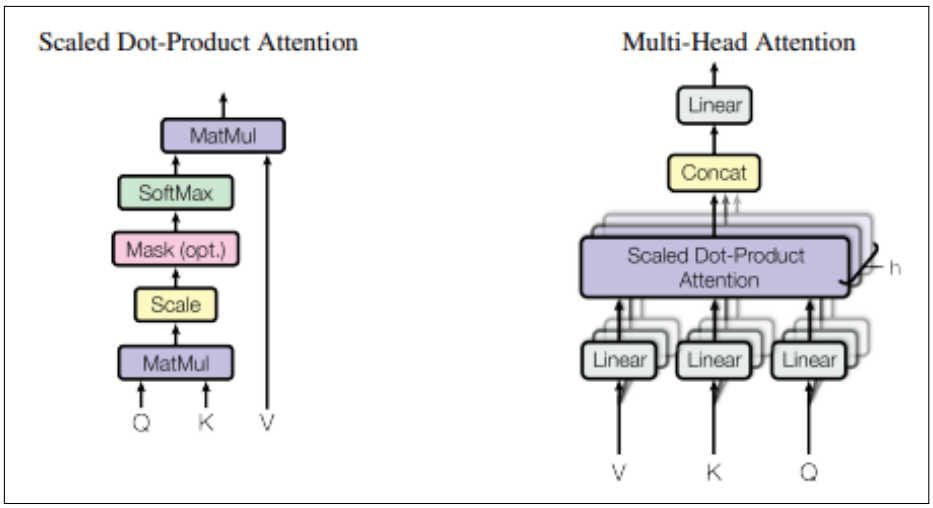
Fonte**:** (VASWANI *et al.*, 2017)

Como já foi dito, o sistema de atenção particular dessa arquitetura é o que faz ela

se destacar. Tal sistema é nomeado como *Scaled Dot-Product Attention*.

Sistemas de atenção, em geral, funcionam como um mapeamento entre uma *querie* e um conjunto de pares chave-valor *(key-value)* para uma saída. Nessa formulação, a *querie*, a chave, o valor e a saída são todos vetores. A saída é calculada como uma soma ponderada dos valores, onde o peso atribuído a cada valor é computado por uma função de compatibilidade entre a *querie* e a chave correspondente. Este conceito permite que o modelo foque dinamicamente nas partes mais relevantes da entrada ao gerar uma saída, em vez de tratar todos os elementos da mesma forma (VASWANI *et al.*, 2017).

Figura 7 – (Esquerda) Atenção escalada por produto escalar e (direita) atenção multi-cabeça .



Fonte**:** (VASWANI *et al.*, 2017)

Especificamente no Transformer, a atenção é implementada como atenção escalada por produto escalar, ou *Scaled Dot-Product Attention* (Figura 7). Dado um conjunto de *querie* Q, chaves K e valores V, todos de dimensão d_k, computa-se o produto escalar entre as *queries* e todas as chaves, divide-se por $\sqrt{d_k}$ para estabilizar os gradientes, e aplica-se uma função *softmax* para obter os pesos de atenção. Em seguida, esses pesos são utilizados para calcular uma média ponderada dos valores. A operação completa é expressa da seguinte forma:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{2.1}$$

Essa forma se aproxima da função de atenção por produto escalar, mas com um ajuste: o fator de escalonamento $\frac{1}{\sqrt{d_k}}$. Na ausência desse fator, quando comparada à função de atenção aditiva, para valores grandes de d_k, os produtos escalares entre as *queries* e as *keys*

tendem a assumir valores de grande magnitude. Isso faz com que os resultados da função *softmax* se aproximem de uma distribuição esparsamente concentrada (quase *one-hot*), o que leva a gradientes extremamente pequenos durante a retropropagação. Como consequência, o aprendizado do modelo se torna ineficiente ou até instável (VASWANI *et al.*, 2017).

Para ampliar ainda mais a capacidade do modelo de capturar diferentes tipos de relacionamentos entre palavras, o *Transformer* utiliza um mecanismo chamado *Multi-Head Attention*. Em vez de calcular uma única atenção, o modelo executa várias operações de atenção em paralelo, cada uma com diferentes projeções lineares de Q, K e V. Os resultados dessas diferentes "cabeças"são então concatenados e projetados novamente, permitindo que o modelo aprenda múltiplas representações contextuais de diferentes subespaços de atenção de forma simultânea.

Esse mecanismo não apenas enriquece a representação contextual de cada palavra na sequência, como também favorece a modelagem de dependências complexas algo essencial para tarefas como tradução, sumarização e geração de texto.

Com essa fundação, o *Transformer* se estabelece como uma arquitetura altamente paralelizável, estável durante o treinamento e eficaz em capturar estruturas linguísticas profundas. Esses fatores explicam seu sucesso generalizado em modelos de linguagem de larga escala, como BERT e GPT.

2.9 MODELOS DE LINGUAGEM DE GRANDE ESCALA (LLMS)

O advento da arquitetura Transformer possibilitou o treinamento em paralelo de modelos de linguagem, graças ao seu mecanismo de atenção baseado em produto escalar. Além disso, essa arquitetura abordou desafios clássicos relacionados à perda ou explosão do gradiente problemas recorrentes em redes profundas ao empregar uma função de ativação como a *softmax* combinada com um fator de escalonamento. Essa solução permitiu que os modelos processassem e aprendessem efetivamente com sequências de texto muito mais longas durante o treinamento, superando uma das principais limitações das arquiteturas anteriores (VASWANI *et al.*, 2017).

Com esse avanço, surgiram os chamados Modelos de Linguagem de Grande Escala, ou Large Language Models (LLMs), que se tornaram o novo paradigma no campo do Processamento de Linguagem Natural (PLN). Os LLMs são modelos estatísticos de linguagem baseados em redes neurais, treinados em larga escala com quantidades massivas de dados textuais. Seu

sucesso recente é fruto de décadas de pesquisa e desenvolvimento em modelagem de linguagem, combinando inovações arquiteturais com avanços significativos em capacidade computacional e disponibilidade de dados (WU *et al.*, 2023; BROWN *et al.*, 2020).

Esses modelos possuem a capacidade de generalizar para uma ampla variedade de tarefas linguísticas com pouco ou nenhum ajuste adicional, o que revolucionou o modo como interagimos com sistemas computacionais baseados em linguagem. Quando comparados aos seus predecessores, por exemplo, os Modelos de Linguagem Estatísticos (SLMs), possuem uma capacidade de generalização muito maior, removendo por inteiro a necessidade de arquiteturas especialistas em tarefas.

Os LLMs mantêm uma estreita relação conceitual com os Modelos de Linguagem Pré-treinados (PLMs), pois ambos seguem o mesmo paradigma de treinamento em duas fases: pré-treinamento e fine-tuning. Nessa abordagem, modelos de linguagem, implementados em arquiteturas Transformer, são inicialmente pré-treinados em datasets gigantescos, compostos por texto não rotulado de escala web. O objetivo dessa fase é capacitar o modelo para tarefas genéricas, como a previsão da próxima palavra em uma sequência ou o preenchimento de lacunas em frases permitindo-lhe aprender um vasto conhecimento linguístico e factual, além de padrões complexos de gramática e semântica (DEVLIN *et al.*, 2018).

Posteriormente, o modelo pré-treinado é ajustado (fine-tuned) para tarefas específicas, adaptando seu conhecimento geral a domínios ou requisitos particulares por meio de um treinamento com dados rotulados mais direcionados (XU *et al.*, 2024).

No entanto, a grande diferença entre os PLMs menores e os LLMs reside fundamentalmente na escala massiva de parâmetros e dados de treinamento dos LLMs, combinada com a eficiência intrínseca da arquitetura Transformer. Essa capacidade de consumir e aprender com volumes sem precedentes de dados se apoia diretamente no paradigma de aprendizado auto-supervisionado.

O treinamento de modelos em uma escala de bilhões ou trilhões de tokens de texto, como é comum para LLMs, seria inviável com abordagens de aprendizado supervisionado tradicional, que exigem que humanos rotulem explicitamente cada exemplo de treinamento. Assim, o aprendizado auto-supervisionado, vem como uma forma de superar essa dificuldade, eliminando a dependência de datasets com dados rotulados. Invés disso, com acesso a uma fonte de dados brutos de escala adequada, sua rtulação é feita automaticamente (BALESTRIERO *et al.*, 2023).

Assim, a disponibilidade massiva de dados e aprendizado semi-autonomo promo-

veram a captura profunda e ampla de padrões linguísticos e de conhecimento de mundo sem precedentes. Diferentemente dos PLMs menores ou dos modelos baseados em RNNs, cuja generalização era mais limitada e frequentemente exigia fine-tuning extensivo para cada nova tarefa, os LLMs exibem capacidades notáveis de aprendizado zero-shot e few-shot. Essas são consideradas habilidades emergentes, pois não são programadas diretamente, mas surgem de forma não linear à medida que a escala do modelo e dos dados de treinamento aumenta (WEI *et al.*, 2022).

- **Aprendizado *Zero-Shot (Zero-Shot Learning)***: Esta capacidade permite que um LLM execute uma tarefa para a qual não foi explicitamente treinado, nem viu nenhum exemplo durante o pré-treinamento ou fine-tuning. O modelo consegue entender a instrução (dada em linguagem natural) e gerar uma resposta apropriada, confiando unicamente no vasto conhecimento e nas representações aprendidas durante sua fase de pré-treinamento massivo (BROWN *et al.*, 2020). Por exemplo, um LLM pode resumir um texto ou gerar um código em uma linguagem específica apenas com base na instrução, sem ter sido treinado em exemplos diretos dessa tarefa.
- **Aprenizado *Few-Shot* (Few-Shot Learning)**: Nesta modalidade, o LLM é capaz de aprender a realizar uma nova tarefa com base em apenas um pequeno número de exemplos (tipicamente de 1 a 5). Ao observar esses poucos exemplos fornecidos no prompt (a instrução de entrada), o modelo infere o padrão ou a intenção da tarefa e aplica esse entendimento a novas entradas. Essa capacidade reduz drasticamente a necessidade de grandes datasets rotulados para fine-tuning em muitas aplicações, acelerando o desenvolvimento e a implantação (BROWN *et al.*, 2020).

Essas capacidades emergentes decorrem diretamente da diversidade e escala dos dados de pré-treinamento, combinadas à flexibilidade da arquitetura Transformer. O mecanismo de autoatenção permite a construção de representações contextuais ricas e eficientes, possibilitando que os modelos desenvolvam uma forma de "raciocínio implícito", frequentemente comparado a um senso comum artificial. Isso representa um avanço expressivo no processamento de linguagem natural e na interação entre humanos e máquinas.

No entanto, à medida que a escala desses modelos aumenta, surgem novos desafios relacionados à infraestrutura necessária para sua execução. Modelos como o GPT-3, LLaMA 3, Gemini e Copilot contam com bilhões de parâmetros, o que demanda uma capacidade computacional significativa tanto para o treinamento quanto para a inferência. Executá-los de maneira eficiente, especialmente em tempo real ou em dispositivos locais, requer soluções especializa-

das.

Para lidar com essas limitações, diversas estratégias têm sido desenvolvidas. Entre elas, destaca-se a quantização, uma técnica que reduz a precisão dos parâmetros do modelo (por exemplo, de 32 para 8 bits), diminuindo o consumo de memória e acelerando o tempo de inferência, com impacto mínimo na performance (ANSAR; GOSWAMI; CHAKRABARTI, 2024). Outras técnicas incluem a poda (pruning), que remove conexões ou neurônios menos importantes, e a destilação de conhecimento (knowledge distillation), que treina um modelo menor para imitar o comportamento de um modelo maior.

Nesse contexto, plataformas como o Ollama surgiram como ferramentas cruciais para democratizar o acesso e o uso eficiente de LLMs. O Ollama permite aos usuários baixar e executar modelos de linguagem de grande escala (incluindo versões quantizadas de modelos populares como Llama 3, Mixtral e Gemma) diretamente em suas máquinas locais, aproveitando ao máximo o hardware disponível. Isso remove a dependência de infraestruturas de nuvem complexas e caras para inferência, facilitando o desenvolvimento de aplicações e a experimentação com LLMs em ambientes privados e controlados. Ao otimizar a execução para CPUs e GPUs de consumidor e ao suportar modelos em formatos mais eficientes, o Ollama exemplifica como as inovações em software e as técnicas de otimização tornam os LLMs mais acessíveis e aplicáveis em uma gama mais ampla de cenários.

2.10 OLLAMA

Apesar do avanço sem precedentes dos Modelos de Linguagem de Grande Escala (LLMs) em capacidade e versatilidade, a execução desses modelos, especialmente em ambientes de desenvolvimento e produção, ainda apresenta desafios significativos. Modelos com bilhões ou trilhões de parâmetros, como GPT-3, LLaMA 3 e Gemini, demandam uma capacidade computacional e de memória substancial tanto para o treinamento quanto para a inferência. Executá-los de maneira eficiente, em tempo real ou em dispositivos locais, requer soluções que otimizem o uso dos recursos de hardware disponíveis (ANSAR; GOSWAMI; CHAKRABARTI, 2024).

Para lidar com essas limitações, diversas estratégias de otimização de modelos têm sido desenvolvidas. Entre as mais proeminentes, destaca-se a quantização, uma técnica que reduz a precisão numérica dos parâmetros do modelo (por exemplo, de 32 bits de ponto flutuante para 8 bits inteiros). Essa redução diminui drasticamente o consumo de memória e acelera o

tempo de inferência, com um impacto frequentemente mínimo na performance e na acurácia do modelo final. Outras técnicas incluem a poda (pruning), que remove conexões ou neurônios menos importantes para reduzir a densidade do modelo, e a destilação de conhecimento (knowledge distillation), que treina um modelo menor (student model) para imitar o comportamento de um modelo maior e mais complexo (teacher model) (LECUN; BENGIO; HINTON, 2015; GOODFELLOW; BENGIO; COURVILLE, 2016).

Nesse contexto de busca por eficiência e acessibilidade, plataformas como o Ollama () surgiram como ferramentas cruciais. O Ollama é uma plataforma de software de código aberto projetada para simplificar a execução e gerenciamento de Modelos de Linguagem de Grande Escala (LLMs) diretamente em máquinas locais, aproveitando ao máximo o hardware do usuário, como CPUs e GPUs de consumidor.

O principal objetivo do Ollama é democratizar o acesso aos LLMs, removendo as barreiras de complexidade e custo associadas à dependência exclusiva de infraestruturas de nuvem ou APIs pagas para inferência. Ele atua como um runtime local que permite aos usuários:

- **Baixar Modelos Pré-treinados:** Oferece um catálogo de LLMs de código aberto populares (como Llama 3, Mixtral, Gemma, entre outros), frequentemente já otimizados por meio de quantização, que podem ser baixados e gerenciados localmente com facilidade.
- **Execução Otimizada:** Gerencia o carregamento e a execução desses modelos, otimizando o uso dos recursos da CPU e, especialmente, da GPU (quando disponível), para proporcionar inferência eficiente mesmo em hardware mais modesto.
- **Interface Simplificada:** Fornece uma interface de linha de comando (CLI) e uma API (Application Programming Interface) fácil de usar, permitindo que desenvolvedores integrem LLMs em suas aplicações de forma direta, sem a necessidade de configurar ambientes complexos ou lidar com bibliotecas de baixo nível.
- **Customização e Fine-tuning Local:** Embora seu foco principal seja a inferência, o Ollama também facilita a experimentação com fine-tuning e a criação de modelos personalizados (modelfiles) a partir de modelos existentes, possibilitando a adaptação para casos de uso específicos.

Ao encapsular as complexidades da execução de LLMs e ao integrar técnicas de otimização como a quantização, o Ollama exemplifica como as inovações em software estão tornando os LLMs mais acessíveis e aplicáveis em uma gama mais ampla de cenários, incluindo o desenvolvimento de soluções personalizadas que operam de forma privada e controlada. Isso é particularmente relevante para aplicações em domínios sensíveis, como o jurídico, onde a

privacidade e o controle sobre os dados são primordiais.

3 TRABALHOS RELACIONADOS

A aplicação de técnicas de Processamento de Linguagem Natural (PLN) e Inteligência Artificial (IA) no âmbito jurídico não é novidade. Há anos, pesquisadores e profissionais do direito exploram como essas tecnologias podem otimizar tarefas, aumentar a eficiência e aprimorar a tomada de decisões nesse setor complexo. Os trabalhos relacionados podem ser categorizados de diversas formas, mas para os propósitos deste estudo, focaremos em abordagens que utilizam IA e PLN para automação de documentos e análise textual no contexto legal, culminando na ascensão dos Modelos de Linguagem de Grande Escala (LLMs).

3.1 AUTOMAÇÃO E ANÁLISE DOCUMENTAL JURÍDICA COM IA E PLN TRADICIONAIS

Historicamente, as iniciativas de inteligência artificial no setor jurídico visavam principalmente a automação de tarefas repetitivas e a análise de grandes volumes de documentos. Ferramentas baseadas em Processamento de Linguagem Natural (PLN) clássico e aprendizado de máquina supervisionado foram desenvolvidas para diversas aplicações, conforme detalhado na literatura (KATZ *et al.*, 2023):

- **Classificação de Documentos**: Categorizar petições, contratos ou decisões judiciais por tipo, assunto ou jurisdição. Isso frequentemente envolvia a extração de características textuais e o uso de classificadores tradicionais.
- **Extração de Informação**: Identificar e extrair entidades específicas de documentos jurídicos, como nomes de partes, datas de prazos processuais, valores de contratos ou cláusulas relevantes.
- **Revisão de Contratos e *Due Diligence***: Ferramentas auxiliavam na revisão de contratos para identificar cláusulas de risco ou inconsistências, acelerando o processo de análise jurídica e due diligence.
- **Previsão de Resultados Judiciais**: Alguns estudos tentaram prever os desfechos de processos judiciais com base em dados históricos, utilizando técnicas de machine learning mais tradicionais.

Embora essas abordagens tenham proporcionado ganhos de eficiência, elas frequentemente apresentavam limitações, como a necessidade de engenharia de características manual, dificuldade em lidar com a ambiguidade inerente à linguagem jurídica e sensibilidade a variações textuais não previstas nas regras ou nos dados de treinamento. Ademais, a capacidade de

generalização para novos domínios ou tipos documentais era restrita, exigindo esforço considerável de adaptação e retreinamento.

3.2 A TRANSFORMAÇÃO COM MODELOS DE LINGUAGEM PRÉ-TREINADOS (PLMS) E LLMS

Ascensão do Deep Learning e, em particular, dos Modelos de Linguagem Pré-Treinados (PLMs) baseados em Transformer (como BERT e GPT), marcou um ponto de inflexão nos trabalhos relacionados em PLN jurídico. Esses modelos trouxeram a capacidade de aprender representações contextuais densas e de alta qualidade para palavras e sentenças, superando as limitações dos embeddings estáticos e das abordagens estatísticas clássicas (DEVLIN *et al.*, 2018; KATZ *et al.*, 2023).

Com o surgimento dos Modelos de Linguagem Pré-Treinados (PLMs), como BERT, GPT e seus derivados, abriram-se novos horizontes para o Processamento de Linguagem Natural (PLN) jurídico, trazendo avanços significativos em várias frentes:

- **Extração e Classificação**: A capacidade desses modelos de capturar nuances contextuais permitiu uma extração de informações e uma classificação de documentos muito mais precisa e robusta, reduzindo a dependência de regras explícitas ou engenharia manual de características.
- **Sistemas de Perguntas e Respostas Jurídicas**: Modelos pré-treinados foram adaptados para responder a perguntas formuladas em linguagem natural com base em bases de conhecimento jurídico, como legislação, jurisprudência e doutrina.
- **Sumarização de Textos Legais**: A criação automática de resumos de sentenças, pareceres e petições, tradicionalmente considerada uma tarefa de alta complexidade devido à densidade informacional e à linguagem formal do texto jurídico, tornou-se mais viável com o uso de PLMs.

A mais recente e significativa evolução nesse cenário é a ascensão dos Modelos de Linguagem de Grande Escala (LLMs). Ao escalar exponencialmente o número de parâmetros e a quantidade de dados de pré-treinamento, os LLMs (como GPT-3, LLaMA e Gemini) transcenderam as capacidades dos PLMs anteriores, desenvolvendo habilidades emergentes como o aprendizado zero-shot e few-shot (BROWN *et al.*, 2020; WEI *et al.*, 2022). Isso significa que um LLM pode realizar uma vasta gama de tarefas jurídicas complexas como gerar rascunhos de documentos, sintetizar longos históricos processuais ou responder a consultas legais complexas

com pouca ou nenhuma necessidade de fine-tuning específico para a tarefa (KATZ *et al.*, 2023).

Trabalhos recentes têm explorado o uso de Modelos de Linguagem de Grande Escala (LLMs) em aplicações jurídicas cada vez mais complexas, com destaque para as seguintes frentes:

- **Geração de Documentos Jurídicos**: Automação da redação de petições simples, contratos padronizados, pareceres introdutórios e até mesmo comunicações jurídicas por e-mail, com linguagem formal e aderente ao contexto normativo.
- **Análise de Contratos em Larga Escala**: Revisão ultrarrápida de contratos com foco em conformidade, identificação de riscos contratuais e detecção de cláusulas-chave, apresentando desempenho superior em relação aos métodos baseados em regras ou aprendizado supervisionado tradicional.
- **Assistentes de Pesquisa Jurídica**: Desenvolvimento de sistemas capazes de responder perguntas jurídicas de forma contextualizada, muitas vezes apresentando fundamentação legal, referências jurisprudenciais ou apontamentos doutrinários relevantes.
- **Sumarização de Audiências e Processos**: Geração de resumos concisos e precisos de audiências, sessões deliberativas ou de todo o histórico de um processo judicial, o que representa um ganho significativo em termos de produtividade e compreensão.

Nesse contexto de avanço, pesquisas atuais também se concentram em garantir a confiabilidade e a explicabilidade das soluções de IA em domínios críticos e sensíveis a erros. A busca pela explicabilidade (XAI) é crucial no domínio jurídico, onde a justificação das conclusões é tão importante quanto a própria conclusão, e os LLMs representam novos desafios e oportunidades nesse sentido (MAHONEY *et al.*, 2019; KATZ *et al.*, 2023).

Nesse cenário, este trabalho se insere na vanguarda da aplicação de LLMs para a automação da elaboração de atas de reunião no contexto jurídico, buscando alavancar as capacidades de compreensão contextual e geração de texto dos LLMs para enfrentar os desafios de precisão, tempo e padronização que ainda persistem na prática manual, ao mesmo tempo em que considera a necessidade de resultados confiáveis e auditáveis.

4 METODOLOGIA

Esta seção descreve os procedimentos desenvolvidos e adotados neste trabalho, com o objetivo de garantir uma implementação eficiente e assegurar a qualidade dos resultados. Inicialmente, apresentam-se os passos relacionados à aquisição, pré-processamento e organização dos dados, visando gerar um *dataset* adequado para testar a aplicação em suas diferentes fases e validar sua eficácia. Em seguida, detalham-se as etapas de desenvolvimento da aplicação, abrangendo sua arquitetura, o fluxo de dados e a integração entre os módulos, com a descrição individual de cada componente.

A metodologia para a definição do *dataset* foi concebida de modo a estabelecer um conjunto de dados capaz de avaliar a qualidade dos resultados gerados pela aplicação e validar sua arquitetura. Para tanto, foi desenvolvido um módulo em Python que realiza todas as etapas necessárias à geração dos dados, desde a aquisição até o refinamento, produzindo um conjunto final que atende aos requisitos mínimos para a validação da aplicação.

Os dados utilizados consistem em tuplas de arquivos PDF e áudios referentes às atas das reuniões. Para possibilitar o processamento e a análise, ambos os tipos de arquivo foram convertidos para o formato de texto: os documentos em PDF foram processados por meio de Reconhecimento Óptico de Caracteres (OCR), técnica utilizada para detectar e converter texto presente em imagens em conteúdo textual editável utilizando o Tesseract, enquanto os arquivos de áudio foram transcritos com o modelo Whisper. Além disso, foram implementados dois módulos adicionais responsáveis pelo pré-processamento dos dados, garantindo a padronização e a limpeza das informações antes da análise.

Após a definição e o pré-processamento do *dataset*, a arquitetura básica do sistema desenvolvido foi projetada para receber transcrições de reuniões e gerar documentos PDF, contendo as informações necessárias dentro do modelo estabelecido. O sistema é composto por um *frontend*, um *backend* e dois módulos principais: um responsável pela lógica de inteligência artificial e outro dedicado à geração do documento PDF.

4.1 COLETA DE DADOS

Para o teste e a validação da aplicação desenvolvida, foi crucial a formação de um *dataset* completo, bem estruturado e alinhado com a problemática central deste trabalho. Nesse contexto, a etapa inicial concentrou-se na identificação de fontes de dados públicas e confiáveis que pudessem fornecer pares documentais específicos: o texto formal da ata e o registro de

como se sucedeu a reunião correspondente, também em texto.

Este conjunto de dados emparelhados é indispensável, não apenas para a validação da aplicação final e a medição de seu desempenho, mas também para a construção do fluxo do sistema em suas fases de processamento. A utilização de dados reais, que espelham o ambiente final de implantação, permitiu o alinhamento e o refinamento dos módulos de aquisição, pré-processamento e análise de forma robusta e representativa.

A instituição selecionada para a extração dos dados foi o Tribunal Regional Eleitoral do Ceará (TRE-CE). Por meio do portal oficial, disponível em <https://www.tre-ce.jus.br/servicos-judiciais/sessoes-de-julgamento/sessoes-plenarias>, foi possível acessar a seção de Serviços Judiciários, especificamente a aba de Sessões, Atas e Pautas de Julgamento. Nessa página, encontram-se os registros das sessões do Plenário, abrangendo o período de junho de 2011 até outubro de 2025.

Para os propósitos deste trabalho, optou-se por extrair dados referentes ao intervalo de março de 2020 até outubro de 2025. Os registros disponibilizados no portal consistem em pares de *links* para cada reunião, sendo essa definida através de sua data. Os pares de *links* consistem em: uma URL que leva direto para o documento oficial no formato PDF, e um *link* para a gravação da respectiva reunião, hospedada no YouTube.

O acesso aos dados foi realizado diretamente por meio do portal oficial do TRE-CE, garantindo a confiabilidade e a legitimidade das informações coletadas. Após a identificação das fontes, foi conduzido um processo de averiguação dos dados disponíveis, verificando a integridade e a consistência entre as atas e as gravações. Em seguida, os dados foram classificados de acordo com sua natureza – documentos PDF e arquivos de áudio.

A complexidade e o volume dos dados a serem processados tornaram impraticável a coleta manual. Para a aquisição em si, foi necessário o desenvolvimento de um módulo de automação em Python capaz de buscar os pares de arquivos dentro do período definido, filtrar os registros válidos, classificá-los baseado na data da reunião e, finalmente, baixar os arquivos para que estivessem prontos para uso no formato esperado pela pipeline que será discutida mais adiante.

Para a implementação desse módulo, foi considerado diversas técnicas, sendo uma delas o *Web Scrapping*. No entanto, visando a robusteza e confiabilidade, foi escolhido uma abordagem mais direta. Investigando o comportamento da página (utilizando ferramentas de desenvolvedor), foi possível deduzir a lógica por trás da construção da URL utilizada para as requisições à API.

O módulo de automação, utilizando a biblioteca *requests* do Python, faz requisições diretas a esses *endpoints* de dados, em vez de processar o HTML da página. Essa abordagem garante que a automação seja mais robusta e resistente a alterações visuais no *frontend* do site, já que ela interage com a camada de dados subjacente. A extração dos metadados em formato estruturado (e.g., JSON) permitiu a classificação e o tratamento rápido das URLs para *download* (PDF) e extração (YouTube).

Adicionalmente ao módulo principal de coleta, foi desenvolvida uma ferramenta dedicada à interação com os dados, motivada pelo volume potencial de arquivos e pela limitação de espaço de armazenamento disponível. Esta ferramenta opera com uma lógica de acesso *on-demand*: os arquivos são baixados localmente somente quando necessários para o processamento e são automaticamente apagados ao fim da iteração, minimizando o consumo de espaço total.

Contudo, para lidar com pequenas amostras e acelerar o desenvolvimento e testes, o módulo também permite o *download* e o armazenamento permanente dos dados localmente.

Especificamente para os arquivos de áudio e vídeo hospedados no YouTube, foi necessário utilizar a biblioteca *YouTubeFix* para interagir com os canais de *streaming* disponíveis. O módulo foi configurado para acessar e extrair apenas a faixa de áudio na melhor qualidade disponível no formato *webm*.

Com isso, foi estabelecido um canal de acesso em tempo real aos dados brutos de interesse para a aplicação. Todos os próximos passos em relação ao desenvolvimento foram feitos a partir desse ponto, aproveitando a disponibilidade e formatação constante dos dados. Para a construção do *dataset* final, esses dados brutos ainda precisariam passar por uma fase de pré-processamento a fim de padronizar as informações no formato de texto.

4.2 PROCESSAMENTO DE DADOS

Esta etapa teve como objetivo a padronização dos dados brutos coletados, ou seja, as atas de reunião em formato PDF e as faixas de áudio no formato *webm*, em uma representação textual estruturada, dequada tanto à aplicação desenvolvida nesse trabalho quanto às técnicas que serão discutidas nos capítulos seguintes, mas mais importante, fiel ao conteúdo original.

A conversão dos dados para formato textual justifica-se, principalmente, por duas razões. Primeiramente, o texto é a modalidade nativa de entrada para a maioria dos modelos de linguagem de larga escala, mesmo que haja, em alguns casos, suporte multimodal. No entanto,

mesmo nesses casos, os modelos costumam realizar internamente uma conversão intermediária entre modalidades, por exemplo de áudio para texto, para assim permitir a compreensão semântica do conetúdo, exposto por Yin *et al.* (2023). Em segundo lugar, como o produto final esperado da pesquisa também é textual, manter o processamento inteiro na mesma modalidade reduz perdas de informação e evita conversões sucessivas, preservando alinhamento semântico entre entrada, processamento e saída.

4.2.1 Processamento de arquivos PDF

Inicialmente, realizou-se uma análise exploratória dos arquivos em formato PDF com o propósito de identificar a estratégia mais adequada para sua conversão em texto, minimizando perdas de conteúdo ou distorções. Dado a natureza do tipo de arquivo (PDF), foi considerado a extração direta do conteúdo, aproveitando a sua estrutura interna.

A abordagem inicial utilizou o módulo *PyMuPDF*, responsável por interpretar a estrutura do PDF, extrair o conteúdo textual e armazená-lo em arquivos de texto simples. Entretanto, a inspeção detalhada do conjunto de dados revelou que uma parcela significativa dos arquivos não continha texto embutido, o que indicava que se tratavam de digitalizações de documentos físicos, assim impossibilitando a extração tradicional.

Diante dessa constatação, tornou-se necessária a aplicação de OCR. Assim, o fluxo de processamento passou, então, a iniciar pela detecção do tipo de documento distinguindo entre PDFs com texto nativo e aqueles compostos por imagens digitalizadas. A classificação define a rota de processamento mais apropriada para cada caso.

Para documentos com texto pesquisável, manteve-se a extração direta via *PyMuPDF*, uma vez que essa abordagem preserva melhor a fidelidade do conteúdo original. Já para os arquivos derivados de material físico, aplicou-se OCR por meio do motor Tesseract. Cada PDF foi previamente segmentado em imagens uma por página e o OCR foi executado em paralelo, página a página. Os resultados da conversão, ao final, foram consolidados em arquivos de texto simples, mantendo uma estrutura uniforme para uso nas etapas posteriores do estudo.

4.2.2 Processamento de Áudio

A conversão do conteúdo das faixas de áudio para a modalidade textual foi, desde o início, enquadrada como uma tarefa apropriada para modelos de inteligência artificial especializados em reconhecimento automático de fala (ASR Automatic Speech Recognition). To-

dos os modelos e ferramentas utilizados são de código aberto e foram executados localmente, assegurando reprodutibilidade, controle total sobre o ambiente de execução e preservação da privacidade dos dados.

Assim como ocorreu com os arquivos em formato PDF, realizou-se inicialmente uma análise exploratória das faixas de áudio extraídas, a fim de compreender suas características e definir um fluxo de processamento adequado ao conjunto de dados.

Durante essa análise, identificaram-se alguns desafios importantes. Um dos primeiros foi a presença de longos períodos de silêncio antes do início e após o término das reuniões, ou seja, períodos que não agragam ao conteúdo final, mas custam no processamento. Além disso, observou-se que as falas dos participantes foram captadas por microfones fixos dispostos sobre as mesas, o que ocasionou variações perceptíveis no volume de voz, uma vez que os interlocutores não mantinham distância constante do microfone.

Outro fator relevante foi a presença de participantes remotos em diversas reuniões. O áudio proveniente dessas intervenções foi registrado de forma indireta captado pelo som emitido pelos alto-falantes da sala e não diretamente do canal de comunicação digital. Essa dupla captação introduziu eco, reverberação e distorções adicionais, afetando consideravelmente a qualidade do sinal.

Por fim, foi identificado um nível muito alto de ruído no ambiente em quase todos os registros das reuniões, normalmente ocasionado pelo microfone de um participante não estar devidamente desligado quando já havia passado a palavra ou outras fontes externas diversas.

Esses fatores evidenciaram a necessidade de um pré-processamento rigoroso, que se tornou uma etapa fundamental da pipeline. A qualidade do áudio processado impacta diretamente o desempenho do modelo de ASR; portanto, diversas medidas foram adotadas para mitigar os problemas identificados.

A principal ferramenta empregada na etapa de transcrição foi o modelo Whisper, da OpenAI, utilizado em sua implementação otimizada faster-whisper. Esse modelo constitui o núcleo do processo de conversão de fala para texto, e todas as etapas anteriores ao seu uso foram projetadas para maximizar sua precisão.

No pré-processamento, uma das primeiras ações consistiu em converter as faixas de áudio para um formato plenamente compatível com o modelo utilizado. Essa conversão foi realizada com a biblioteca FFmpeg, uma ferramenta amplamente utilizada para manipulação, transcodificação e análise de áudio e vídeo.

Os arquivos originais, fornecidos em formato WebM, foram reamostrados para uma taxa de 16 kHz, convertidos para canal único (mono) e codificado para o padrão PCM de 16 bits. A adoção desses parâmetros, além de ser o recomendado para ser processado pelo modelo Whisper, visa assegurar a fidelidade do sinal.

Durante o processo de conversão, foram aplicados filtros voltados à melhoria da inteligibilidade do áudio e à redução de variações decorrentes das condições reais de gravação. O primeiro deles foi o *dynaudnorm* (*Dynamic Audio Normalizer*), responsável por ajustar dinamicamente o ganho do sinal ao longo do tempo. Esse filtro analisa janelas sucessivas do áudio e aplica amplificação apenas quando necessário, respeitando um limite máximo de ganho, neste caso, 15 dB, o que impede saturação e preserva a integridade do sinal. Assim, trechos de baixa intensidade são reforçados, enquanto partes mais intensas têm sua amplificação controlada, resultando em um áudio mais uniforme e adequado para processamento posterior.

Em complemento, utilizou-se o filtro acompressor, cuja função consiste em atenuar picos abruptos de amplitude e promover maior uniformidade no sinal. A aplicação combinada desses filtros contribuiu para mitigar discrepâncias de volume e aprimorar a clareza acústica do material processado.

O FFmpeg foi configurado para produzir a saída diretamente por meio de fluxo binário (pipe), eliminando a necessidade de criação de arquivos intermediários no sistema de armazenamento. O áudio resultante foi armazenado em um objeto de memória volátil, permitindo seu encaminhamento imediato às etapas subsequentes da pipeline de transcrição. Esse procedimento reduziu o tempo total de processamento, simplificou o fluxo operacional e forneceu ao modelo Whisper um sinal mais consistente e menos suscetível a ruídos e variações indesejadas, favorecendo, assim, a precisão dos resultados obtidos.

Após a conversão e padronização inicial das faixas de áudio, procedeu-se à etapa de redução de ruído, cuja finalidade consistiu em mitigar interferências acústicas presentes nas gravações originais. Para esse fim, empregou-se um modelo de rede neural recorrente (RNN) especialmente treinado para a tarefa de supressão de ruído, disponibilizado por GregorR (2018) em repositório público. O modelo, no formato .rnnn, foi executado localmente por meio do filtro arnndn do FFmpeg, que implementa uma interface nativa para modelos de denoising baseados em redes neurais.

O áudio, previamente convertido e normalizado, foi encaminhado ao FFmpeg por meio de fluxo binário, evitando a criação de arquivos intermediários. Durante o processamento, aplicou-se o filtro arnndn, configurado para utilizar o modelo selecionado, o qual é especiali-

zado na redução de ruído estacionário e de baixa frequência, característico de ambientes com microfones abertos, ventilação ambiente ou reverberações persistentes.

Em complemento, aplicou-se o módulo loudnorm com o objetivo de promover a normalização do nível sonoro percebido, reduzir o intervalo dinâmico e assegurar limites máximos de pico adequados às condições típicas de fala. Essa combinação de filtros resultou em um sinal mais uniforme, com redução significativa de ruídos de fundo e melhora na clareza das vozes registradas.

O resultado do processo foi gerado em formato WAV, com codificação PCM de 16 bits e taxa de amostragem de 16 kHz, preservando a compatibilidade com as etapas subsequentes da pipeline de transcrição. O áudio processado foi mantido em memória volátil e reintegrado ao fluxo operacional sem necessidade de gravação em disco. A etapa de redução de ruído mostrou-se especialmente relevante devido à presença recorrente de interferências ambientais nas gravações das reuniões, impactando diretamente a acurácia do modelo de ASR utilizado.

Após a etapa de redução de ruído, iniciou-se o processo de diarização de falantes, cujo objetivo consiste em identificar automaticamente os segmentos de fala presentes no áudio e atribuí-los aos respectivos interlocutores. Para essa finalidade, empregou-se o modelo Pyannote (PLAQUET; BREDIN, 2023), amplamente reconhecido na literatura por seu elevado desempenho em ambientes heterogêneos de gravação e por utilizar arquiteturas de deep learning especializadas na detecção de mudanças de locutor em cenários de fala contínua.

Antes da adoção do modelo de diarização, avaliou-se o uso de um filtro do tipo Voice Activity Detection (VAD) para eliminar trechos sem atividade vocal e reduzir o volume de dados processados. Entretanto, essa abordagem mostrou-se contraproducente: além de introduzir uma etapa adicional e aumentar o custo computacional, a remoção de períodos de silêncio prejudicou o desempenho da diarização, uma vez que o Pyannote utiliza justamente a distribuição temporal das pausas como parte do contexto necessário para identificar transições entre falantes. Assim, optou-se por fornecer ao modelo o áudio integral, preservando os intervalos de inatividade.

Seguindo o fluxo estabelecido na pipeline, o áudio previamente normalizado e tratado foi encaminhado ao modelo por meio de fluxo binário em memória, evitando operações de escrita e leitura em disco e garantindo maior eficiência na execução. A segmentação em falas do áuido feita pelo modelo consiste na divisão sequencial quanto ao tempo, onde cada um representa um intervalo contínuo de fala idnetificado pelo algoritmo.

Cada segmento é composto por três informações principais: o instante de início (*startt*), o instante de término (*end*) e o indentificador de falante (*speaker*). Esta última informação serve apenas como uma forma de diferenciar os interlocutores, permitindo distinguir quantas vozes distintas foram detectadas ao longo do áudio.

Ainda na fase de diarização, foi preciso realizar um pós processamento em cima dos segmentos extraídos pelo pynnote, destinado a corrigir fragmentações de segmentos contínuos e a indentificação errônea da identidade do locutor. Embora o modelo seja capaz de identificar com elevada precisão os trechos de fala e seus respectivos locutores, é comum que pequenas variações acústicas produzam divisões indevidas dentro de uma mesma fala contínua, resultando em múltiplos segmentos muito curtos e adjacentes atribuídos ao mesmo falante.

Para lidar como esse problema, desenvolveu-se um algoritmo de unificação (*merge*) de segmentos baseada na análise estatística da pausas do trechos consecutivos de um mesmo locutor identificado. Inicialmente, os seguimentos são agrupados em vetores respectivos ao seu falante e em seguida são ordenados temporalmente. Com isso, para cada grupo, calcula-se a diferença entre o fim e o iníco dos segmentos. Essas diferenças, representativas da duração das pausas intralocutor, são reunidas em um único vetor e filtradas para remover valores negativos ou não significativos.

Com esse conjunto de intervalos, aplica-se o método *Multi-Otsu Thresholding*, que realiza particionamento automatizado dos valores em, para essa implementação, três faixas, identificando limiares que separam pausas curtas, intermediárias e longas. O menor desses limiares é então usado como *threshold* para filtrar as pausas artificiais consecutivas entre segmentos para, assim, realizar o processo de fusão.

4.3 MÉTRICAS DE AVALIAÇÃO

5 RESULTADOS

6 CONCLUSÕES E TRABALHOS FUTUROS

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris. Nullam eleifend justo in nisl. In hac habitasse platea dictumst. Morbi nonummy. Aliquam ut felis. In velit leo, dictum vitae, posuere id, vulputate nec, ante. Maecenas vitae pede nec dui dignissim suscipit. Morbi magna. Vestibulum id purus eget velit laoreet laoreet. Praesent sed leo vel nibh convallis blandit. Ut rutrum. Donec nibh. Donec interdum. Fusce sed pede sit amet elit rhoncus ultrices. Nullam at enim vitae pede vehicula iaculis.

6.1 CONTRIBUIÇÕES DO TRABALHO

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

6.2 LIMITAÇÕES

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

6.3 TRABALHOS FUTUROS

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

REFERÊNCIAS

ANSAR, W.; GOSWAMI, S.; CHAKRABARTI, A. **A Survey on Transformers in NLP with Focus on Efficiency**. arXiv, 2024. Disponível em: <https://arxiv.org/abs/2406.16893>.

BALESTRIERO, R.; IBRAHIM, M.; SOBAL, V.; MORCOS, A.; SHEKHAR, S.; GOLDSTEIN, T.; BORDES, F.; BARDES, A.; MIALON, G.; TIAN, Y.; SCHWARZSCHILD, A.; WILSON, A. G.; GEIPING, J.; GARRIDO, Q.; FERNANDEZ, P.; BAR, A.; PIRSIAVASH, H.; LECUN, Y.; GOLDBLUM, M. **A Cookbook of Self-Supervised Learning**. arXiv, 2023. Disponível em: <https://arxiv.org/abs/2304.12210>.

BODEN, M. A. **Artificial intelligence: a very short introduction**. [Oxford]: Oxford University Press, 2018. (Very short introductions, 575). OCLC: 1050938367. ISBN 9780191821448.

BRASIL. **Lei nž 8.159, de 8 de janeiro de 1991**. 1991. Disponível em: <https://www.planalto.gov.br/ccivil_03/leis/l8159.htm>.

BRASIL. **LEI COMPLEMENTAR Nž 131, DE 27 DE MAIO DE 2009**. 2009. Disponível em: <https://www.planalto.gov.br/ccivil_03/leis/lcp/lcp131.htm>.

BRASIL. **LEI Nž 12.527, DE 18 DE NOVEMBRO DE 2011**. 2011. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm>.

BROWN, T. B.; MANN, B.; RYDER, N.; SUBBIAH, M.; KAPLAN, J.; DHARIWAL, P.; NEELAKANTAN, A.; SHYAM, P.; SASTRY, G.; ASKELL, A.; AGARWAL, S.; HERBERT-VOSS, A.; KRUEGER, G.; HENIGHAN, T.; CHILD, R.; RAMESH, A.; ZIEGLER, D. M.; WU, J.; WINTER, C.; HESSE, C.; CHEN, M.; SIGLER, E.; LITWIN, M.; GRAY, S.; CHESS, B.; CLARK, J.; BERNER, C.; MCCANDLISH, S.; RADFORD, A.; SUTSKEVER, I.; AMODEI, D. **Language Models are Few-Shot Learners**. arXiv, 2020. Disponível em: <https://arxiv.org/abs/2005.14165>.

CHO, K.; MERRIENBOER, B. van; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H.; BENGIO, Y. **Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation**. arXiv, 2014. Disponível em: <https://arxiv.org/abs/1406.1078>.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. arXiv, 2018. Disponível em: <https://arxiv.org/abs/1810.04805>.

FERNANDES, L. C. D. C. PROPAGANDA, TRANSPARÊNCIA E ACCOUNTABILITY: A CONSTRUÇÃO DE INDICADORES PARA UMA GOVERNANÇA DEMOCRÁTICA. **Revista Panorama - Revista de Comunicação Social**, v. 11, n. 1, p. 46, set. 2021. ISSN 2237-1087. Disponível em: <http://seer.pucgoias.edu.br/index.php/panorama/article/view/9026>.

FERREIRA, E. D.; CAMBRUSSI, M. F. Redação Oficial. 2015. Disponível em: <http://educapes.capes.gov.br/handle/capes/145384>.

FU, Z.; LAM, W.; YU, Q.; SO, A. M.-C.; HU, S.; LIU, Z.; COLLIER, N. **Decoder-Only or Encoder-Decoder? Interpreting Language Model as a Regularized Encoder-Decoder**. arXiv, 2023. Disponível em: <https://arxiv.org/abs/2304.04052>.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. Cambridge, Massachusetts: The MIT Press, 2016. (Adaptive computation and machine learning). ISBN 9780262035613.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. H. **The elements of statistical learning: data mining, inference, and prediction**. 2nd ed. ed. New York, NY: Springer, 2009. (Springer series in statistics). ISBN 9780387848570 9780387848587.

HOCHREITER, S.; SCHMIDHUBER, J. Long Short-Term Memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, nov. 1997. ISSN 0899-7667, 1530-888X. Disponível em: <https://direct.mit.edu/neco/article/9/8/1735-1780/6109>.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural Networks**, v. 2, n. 5, p. 359–366, jan. 1989. ISSN 08936080. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/0893608089900208>.

JURAFSKY, D.; MARTIN, J. H. **Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition**. 2. ed. [nachdr.]. ed. Upper Saddle River, NJ: Prentice Hall, 2009. ISBN 9780131873216.

KANDEL, E. R. (Ed.). **Principles of neural science**. 5th ed. ed. New York: McGraw-Hill, 2013. ISBN 9780071390118.

KATZ, D. M.; HARTUNG, D.; GERLACH, L.; JANA, A.; II, M. J. B. **Natural Language Processing in the Legal Domain**. arXiv, 2023. ArXiv:2302.12039. Disponível em: <http://arxiv.org/abs/2302.12039>.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet classification with deep convolutional neural networks. **Communications of the ACM**, v. 60, n. 6, p. 84–90, maio 2017. ISSN 0001-0782, 1557-7317. Disponível em: <https://dl.acm.org/doi/10.1145/3065386>.

LE, P.; ZUIDEMA, W. **Quantifying the vanishing gradient and long distance dependency problem in recursive neural networks and recursive LSTMs**. arXiv, 2016. ArXiv:1603.00423. Disponível em: <http://arxiv.org/abs/1603.00423>.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, v. 521, n. 7553, p. 436–444, maio 2015. ISSN 0028-0836, 1476-4687. Disponível em: <https://www.nature.com/articles/nature14539>.

MAHONEY, C. J.; ZHANG, J.; HUBER-FLIFLET, N.; GRONVALL, P.; ZHAO, H. **A Framework for Explainable Text Classification in Legal Document Review**. arXiv, 2019. Disponível em: <https://arxiv.org/abs/1912.09501>.

MCCARTHY, J.; MINSKY, M.; ROCHESTER, N.; SHANNON, C. **A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence**. [S.l.], 1955.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, v. 5, n. 4, p. 115–133, dez. 1943. ISSN 0007-4985, 1522-9602. Disponível em: <http://link.springer.com/10.1007/BF02478259>.

MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G.; DEAN, J. **Distributed Representations of Words and Phrases and their Compositionality**. arXiv, 2013. ArXiv:1310.4546 version: 1. Disponível em: <http://arxiv.org/abs/1310.4546>.

MITCHELL, T. M. **Machine learning**. Nachdr. New York: McGraw-Hill, 2013. (McGraw-Hill series in Computer Science). ISBN 9780070428072 9780071154673.

MORAVEC, H. **Mind children: the future of robot and human intelligence**. 4. print. ed. Cambridge: Harvard Univ. Press, 1995. ISBN 9780674576186.

NADKARNI, P. M.; OHNO-MACHADO, L.; CHAPMAN, W. W. Natural language processing: an introduction. **Journal of the American Medical Informatics Association**, v. 18, n. 5, p. 544–551, set. 2011. ISSN 1067-5027, 1527-974X. Disponível em: <https://academic.oup.com/jamia/article-lookup/doi/10.1136/amiajnl-2011-000464>.

OLLAMA. **Ollama**. Disponível em: <https://ollama.com/>.

PENNINGTON, J.; SOCHER, R.; MANNING, C. Glove: Global Vectors for Word Representation. In: **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. Doha, Qatar: Association for Computational Linguistics, 2014. p. 1532–1543. Disponível em: <http://aclweb.org/anthology/D14-1162>.

PLAQUET, A.; BREDIN, H. Powerset multi-class cross entropy loss for neural speaker diarization. In: **Proc. INTERSPEECH 2023**. [S.l.: s.n.], 2023.

REPúBLICA, P. d. **Manual de redação da Presidência da República**. [S.l.]: Presidência da República, 2018. ISBN 9788585142964.

REZENDE, S. O. (Ed.). **Sistemas inteligentes: fundamentos e aplicações**. 1. ed. ed. Barueri, SP: Ed. Manole, 2003. ISBN 9788520416839.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, n. 6, p. 386–408, 1958. ISSN 1939-1471, 0033-295X. Disponível em: <https://doi.apa.org/doi/10.1037/h0042519>.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, v. 323, p. 533–536, 1986. Disponível em: <https://api.semanticscholar.org/CorpusID:205001834>.

RUSSELL, S. J.; NORVIG, P. **Artificial intelligence: a modern approach**. Third edition, global edition. Boston Columbus Indianapolis: Pearson, 2016. (Prentice Hall series in artificial intelligence). ISBN 9780136042594 9781292153971.

SCHMIDT, R. M. **Recurrent Neural Networks (RNNs): A gentle Introduction and Overview**. arXiv, 2019. ArXiv:1912.05911. Disponível em: <http://arxiv.org/abs/1912.05911>.

SHERSTINSKY, A. **Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network**. arXiv, 2023. ArXiv:1808.03314. Disponível em: <http://arxiv.org/abs/1808.03314>.

TOPAL, M. O.; BAS, A.; HEERDEN, I. v. **Exploring Transformers in Natural Language Generation: GPT, BERT, and XLNet**. arXiv, 2021. ArXiv:2102.08036. Disponível em: <http://arxiv.org/abs/2102.08036>.

TURING, A. M. I.COMPUTING MACHINERY AND INTELLIGENCE. **Mind**, LIX, n. 236, p. 433–460, out. 1950. ISSN 1460-2113, 0026-4423. Disponível em: <https://academic.oup.com/mind/article/LIX/236/433/986238>.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L.; POLOSUKHIN, I. **Attention Is All You Need**. arXiv, 2017. Disponível em: <https://arxiv.org/abs/1706.03762>.

VENNERØD, C. B.; KJæRRAN, A.; BUGGE, E. S. **Long Short-term Memory RNN**. arXiv, 2021. ArXiv:2105.06756. Disponível em: <http://arxiv.org/abs/2105.06756>.

WEI, J.; TAY, Y.; BOMMASANI, R.; RAFFEL, C.; ZOPH, B.; BORGEAUD, S.; YOGATAMA, D.; BOSMA, M.; ZHOU, D.; METZLER, D.; CHI, E. H.; HASHIMOTO, T.; VINYALS, O.; LIANG, P.; DEAN, J.; FEDUS, W. **Emergent Abilities of Large Language Models**. arXiv, 2022. Disponível em: <https://arxiv.org/abs/2206.07682>.

WEISS, G. (Ed.). **Multiagent systems: a modern approach to distributed artificial intelligence**. 3. print. ed. Cambridge, Mass.: MIT Press, 2001. ISBN 9780262731317 9780262232036.

WU, J.; GAN, W.; CHEN, Z.; WAN, S.; YU, P. S. Multimodal Large Language Models: A Survey. In: **2023 IEEE International Conference on Big Data (BigData)**. Sorrento, Italy: IEEE, 2023. p. 2247–2256. ISBN 9798350324457. Disponível em: <https://ieeexplore.ieee.org/document/10386743/>.

XU, H.; SHARAF, A.; CHEN, Y.; TAN, W.; SHEN, L.; DURME, B. V.; MURRAY, K.; KIM, Y. J. **Contrastive Preference Optimization: Pushing the Boundaries of LLM Performance in Machine Translation**. arXiv, 2024. Disponível em: <https://arxiv.org/abs/2401.08417>.

YIN, S. *et al.* A survey on multimodal large language models. **arXiv preprint arXiv:2306.13549**, 2023. Disponível em: <https://arxiv.org/pdf/2306.13549>. Acesso em: 8 nov. 2025. Disponível em: <https://arxiv.org/pdf/2306.13549>.

APÊNDICES

APÊNDICE A – Lorem Ipsum

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

APÊNDICE B – Modelo de Capa

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

APÊNDICE C – Termo de Fiel Depositário

Pesquisa: ANÁLISE DA MORTALIDADE INFANTIL COM MALFORMAÇÕES CONGÊNITAS.

Pelo presente instrumento que atende às exigências legais, a Sra. Maria Consuelo Martins Saraiva, "fiel depositário" com o cargo de Secretária Municipal de Saúde de Iracema, após ter tomado conhecimento do protocolo de pesquisa intitulado: ANÁLISE DA MORTALIDADE INFANTIL COM MALFORMAÇÕES CONGÊNITAS. Analisando a repercussão desse estudo no contexto da saúde pública e epidemiologia, autoriza Karla Maria da Silva Lima, enfermeira, aluna do Curso de Mestrado Acadêmico em Enfermagem da Universidade Estadual do Ceará (UECE), sob orientação do Prof. Dr. José Maria de Castro, da UECE, ter acesso aos bancos de dados do Sistema de Informação sobre Nascidos Vivos e do Sistema de Informação sobre Mortalidade da Secretaria Municipal de Saúde de Iracema, objeto deste estudo, e que se encontram sob sua total responsabilidade. Fica claro que o Fiel Depositário pode a qualquer momento retirar sua AUTORIZAÇÃO e ciente de que todas as informações prestadas tornar-se-ão confidenciais e guardadas por força de sigilo profissional, assegurando que os dados obtidos da pesquisa serão somente utilizados para estudo.

ANEXOS

ANEXO A – Exemplo de Anexo

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

ANEXO B – Dinâmica das classes sociais

Etiam ac leo a risus tristique nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus. Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor.