



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

RODRIGO SILVA PERES

**GRANDES MODELOS DE LINGUAGEM NA RESOLUÇÃO DE QUESTÕES DE
VESTIBULAR: O CASO DOS INSTITUTOS MILITARES BRASILEIROS**

RIO DE JANEIRO

2023

RODRIGO SILVA PERES

**GRANDES MODELOS DE LINGUAGEM NA RESOLUÇÃO DE QUESTÕES DE
VESTIBULAR: O CASO DOS INSTITUTOS MILITARES BRASILEIROS**

Dissertação apresentada como requisito parcial para
obtenção do título de Mestre pelo Programa de Pós-
Graduação em Informática da Universidade Federal
do Estado do Rio de Janeiro (UNIRIO)

Orientador: Carlos Eduardo Ribeiro de Mello

Coorientadora: Laura de Oliveira Fernandes Moraes

RIO DE JANEIRO

2023

Catálogo informatizada pelo(a) autor(a)

P437g Peres, Rodrigo Silva
Grandes Modelos de Linguagem na resolução de
questões de vestibular: o caso dos Institutos
Militares brasileiros / Rodrigo Silva Peres. -- Rio
de Janeiro, 2023.
74 f.

Orientador: Carlos Eduardo Ribeiro de Mello.
Coorientadora: Laura de Oliveira Fernandes
Moraes.

Dissertação (Mestrado) - Universidade Federal do
Estado do Rio de Janeiro, Programa de Pós-Graduação
em Informática, 2023.

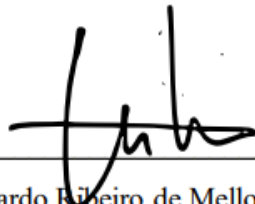
1. Large Language Models. 2. Engenharia de
prompt. 3. Processamento de Linguagem Natural. I.
Mello, Carlos Eduardo Ribeiro de, orient. II.
Moraes, Laura de Oliveira Fernandes, coorient. III.
Título.

RODRIGO SILVA PERES

GRANDES MODELOS DE LINGUAGEM NA RESOLUÇÃO DE QUESTÕES DE VESTIBULAR: O CASO DOS INSTITUTOS MILITARES BRASILEIROS

Dissertação apresentada como requisito parcial para obtenção do título de Mestre pelo Programa de Pós-Graduação em Informática da Universidade Federal do Estado do Rio de Janeiro (UNIRIO)

Aprovada por:



Carlos Eduardo Ribeiro de Mello, Ph.D. (Orientador)

Universidade Federal do Estado do Rio de Janeiro – UNIRIO



Laura de Oliveira Fernandes Moraes, D.Sc. (Coorientadora)


Universidade Federal do Estado do Rio de Janeiro – UNIRIO



Sean Wolfgang Matsui Siqueira, D.Sc.

Universidade Federal do Estado do Rio de Janeiro – UNIRIO

DocuSigned by:



4B8111BD75104B5...

Rodrigo Frassetto Nogueira, Ph.D.

Universidade Estadual de Campinas - UNICAMP

RIO DE JANEIRO, RJ – BRASIL

AGOSTO DE 2023

Dedico este trabalho à minha família e aos meus amigos, em especial minha mãe Cláudia, meu pai Eduardo e minha irmã Renata.

AGRADECIMENTOS

À toda minha família, em especial minha mãe Cláudia, meu pai Eduardo e minha irmã Renata, que souberem entender minhas ausências, aguentaram meu mau humor e cujo afeto e apoio me motivaram nessa caminhada universitária.

Agradeço a Deus por tudo. Pelas oportunidades, desafios, derrotas e vitórias. Agradeço por ter me guiado e me dado força ao longo do caminho para que eu continuasse persistindo até o fim.

Agradeço ao Programa de Pós-Graduação em Informática (PPGI) da Universidade Federal do Estado do Rio de Janeiro (UNIRIO) e aos seus professores pelos ensinamentos e trocas. Aproveito para estender os agradecimentos aos servidores técnico-administrativos, o suporte dado foi fundamental durante o curso.

Seguindo com os agradecimentos, gostaria de deixar um obrigado especial aos meus professores e orientadores Carlos Eduardo Mello e Laura Moraes por acreditarem no meu potencial, pelo tempo dedicado e pelas discussões sempre enriquecedoras, e aos membros da banca, professores Sean Siqueira e Rodrigo Nogueira, por terem aceitado o convite e por terem se disposto a avaliar meu trabalho.

Por último, mas não menos importante, gostaria de agradecer aos meus amigos, tanto os de longa data quanto aos que conheci no curso. Obrigado pela companhia, pelas conversas, pelas risadas, pelas parcerias, enfim, obrigado por terem tornado tudo bem mais fácil.

PERES, Rodrigo Silva. **Grandes Modelos de Linguagem na resolução de questões de vestibular**: o caso dos institutos militares brasileiros. UNIRIO, 2023. 74 f. Dissertação (Mestrado em Informática) - Departamento de Informática Aplicada, Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, 2023.

RESUMO

Esta pesquisa analisou o desempenho dos Grandes Modelos de Linguagem ou *Large Language Models* (LLMs) combinados com técnicas de *in-context learning* na resolução de questões complexas de exames vestibulares escritas em português. O conjunto de dados utilizado foi construído utilizando questões de exames vestibulares de duas prestigiadas instituições militares brasileiras, o Instituto Militar de Engenharia (IME) e o Instituto Tecnológico da Aeronáutica (ITA). Foram conduzidos experimentos controlados avaliando diferentes métodos de *prompt* com os modelos text-davinci-003, GPT-3.5-turbo e GPT-4 para verificar seu desempenho ao fornecer as respostas. Os resultados mostraram que os LLMs recentes otimizados para conversação alcançaram melhores métricas, especialmente quando utilizam técnicas de *in-context learning*, como *chain-of-thought* (CoT). O GPT-4 alcançou uma acurácia geral de 55% usando essa abordagem, superando os outros modelos. Analisando o desempenho por disciplina, obteve-se melhores resultados nas questões de Português, enquanto o destaque negativo foi para a Matemática. As questões do ITA foram ligeiramente menos desafiadoras para os LLMs do que as do IME, sugerindo menor complexidade para resolução por parte dos modelos. A pesquisa aponta o potencial dos Grandes Modelos de Linguagem na resolução de problemas complexos, fornecendo evidências valiosas para o aprimoramento e a adequada implementação de técnicas de Processamento de Linguagem Natural em contextos educacionais. Outros aspectos podem ser explorados posteriormente como uma variedade maior de modelos, utilização de técnicas de *prompt* de múltiplos turnos, treinamento de modelos com mais dados em português e a investigação do impacto do formato das questões.

Palavras-chave: Grandes Modelos de Linguagem. Engenharia de prompt. Processamento de Linguagem Natural.

PERES, Rodrigo Silva. **Large Language Models in solving college entrance exam questions: the case of the brazilian military institutes.** UNIRIO, 2023. 74 f. Dissertação (Mestrado em Informática) – Departamento de Informática Aplicada, Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, 2023.

ABSTRACT

This research analyzed the performance of Large Language Models (LLMs) combined with in-context learning techniques in solving complex university entrance exam questions written in Portuguese. The dataset used was constructed using questions from entrance exams from two prestigious Brazilian military institutions, Military Institute of Engineering (IME) and Technological Institute of Aeronautics (ITA). Controlled experiments were conducted evaluating different prompt methods with the text-davinci-003, GPT-3.5-turbo and GPT-4 models to verify their performance in providing answers. The results showed that recent LLMs optimized for conversation achieved better metrics, especially when using in-context learning techniques such as chain-of-thought (CoT). GPT-4 achieved an overall accuracy of 55% using this approach, outperforming other models. Analyzing performance by subject, better results were obtained in Portuguese questions, while the negative highlight was for Mathematics. The ITA questions were slightly less challenging for the LLMs than the IME ones, suggesting less complexity for the models to solve. The research highlights the potential of Large Language Models in solving complex problems, providing valuable evidence for the improvement and adequate implementation of Natural Language Processing techniques in educational contexts. Other aspects can be explored later such as a greater variety of models, use of multiple turn prompt techniques, training models with more data in Portuguese and investigating the impact of question format.

Keywords: Large Language Models. Prompt Engineering. Natural Language Processing.

LISTA DE ILUSTRAÇÕES

Figura 1	Relações da área de <i>Deep Learning</i>	18
Figura 2	Comparação entre <i>Machine Learning</i> em relação a <i>Deep Learning</i>	18
Figura 3	Exemplo da arquitetura RNN.....	19
Figura 4	Bloco LSTM.....	20
Figura 5	Representação da Bi-LSTM.....	21
Figura 6	Arquitetura genérica de CNN.....	22
Figura 7	Arquitetura do modelo <i>Transformer</i>	25
Figura 8	Árvore evolutiva dos Grandes Modelos de Linguagem.....	27
Figura 9	Fórmulas da Lei de escala KM.....	28
Figura 10	Funcionamento dos <i>prompts</i> de um e múltiplos estágios.....	31
Figura 11	Técnicas das técnicas <i>Zero-shot</i> , <i>One-shot</i> e <i>Few-shot</i>	32
Figura 12	Diferença entre <i>One-shot</i> e <i>Chain-of-Thought</i>	34
Figura 13	Ilustração da técnica de <i>Zero-shot CoT</i>	36
Figura 14	Comparação entre <i>Zero-shot CoT</i> e <i>Plan-and-Solve</i>	37
Figura 15	Número de questões por disciplina.....	46
Figura 16	Modelos da Família <i>Transformers</i>	48
Quadro 1	Informações sobre os LLMs selecionados.....	49
Figura 17	Ilustração do funcionamento do software.....	51
Figura 18	Quantidade de questões certas por Disciplina.....	62
Quadro 2	Questão de Português com todas as respostas corretas.....	64
Quadro 3	Questão de Química com todas as respostas corretas.....	65
Quadro 4	Questão de Português com todas as respostas incorretas.....	66
Quadro 5	Questão de Matemática com todas as respostas incorretas.....	67

LISTA DE TABELAS

Tabela 1	Desempenho dos LLMs em diferentes testes aplicados nos EUA.....	39
Tabela 2	Resultados do JEE por área do conhecimento.....	41
Tabela 3	Resultados do ENEM por área do conhecimento.....	42
Tabela 4	Distribuição das questões por prova e ano.....	45
Tabela 5	Resultados gerais por modelo e técnica de engenharia de prompt.....	54
Tabela 6	Resultados por disciplina.....	57
Tabela 7	Resultados por ano.....	60
Tabela 8	Resultados por prova.....	61

LISTA DE ABREVIATURAS E SIGLAS

AMTCL	<i>Association for Machine Translation and Computational Linguistics</i>
Bi-LSTM	<i>Bidirectional Long Short-Term Memory</i>
CoT	<i>Chain-of-Thought</i>
CNN	<i>Convolutional Neural Network</i>
DL	<i>Deep Learning</i>
ENEM	Exame Nacional do Ensino Médio
GRU	<i>Gated Recurrent Unit</i>
IES	Instituições de Ensino Superior
IME	Instituto Militar de Engenharia
ITA	Instituto Tecnológica da Aeronáutica
JEE	<i>Joint Entrance Examination</i>
LLMs	<i>Large Language Models</i>
LSAT	<i>Law School Admission Test</i>
LSTM	<i>Long Short-Term Memory</i>
NLP	<i>Natural Language Processing</i>
PLN	Processamento de Linguagem Natural
RNN	<i>Recurrent Neural Network</i>
SAT	<i>Scholastic Assessment Test</i>
Seq2Seq	<i>Sequence-to-Sequence</i>

SUMÁRIO

1	INTRODUÇÃO.....	13
1.1	PROBLEMA.....	14
1.2	JUSTIFICATIVA.....	14
1.3	OBJETIVOS.....	15
1.3.1	Objetivo geral.....	15
1.3.2	Objetivos específicos.....	15
1.4	ORGANIZAÇÃO DO TRABALHO.....	16
2	REFERENCIAL TEÓRICO.....	17
2.1	<i>DEEP LEARNING</i> APLICADO A PROBLEMAS DE PLN.....	17
2.2	ATENÇÃO, ARQUITETURA <i>TRANSFORMER</i> E OS LLM.....	23
2.3	OS PARADIGMAS DOS MODELOS DE LINGUAGEM.....	29
2.4	<i>IN-CONTEXT LEARNING</i>	30
2.4.1	<i>Zero-shot e Few-shot</i>.....	31
2.4.2	<i>Chain-of-Thought</i>.....	33
2.4.3	<i>Zero-shot Chain-of-Thought</i>.....	35
2.4.4	<i>Plan-and-Solve</i>.....	37
3	TRABALHOS RELACIONADOS.....	39
4	METODOLOGIA.....	44
4.1	COLETA DE DADOS.....	44
4.2	SELEÇÃO DOS MODELOS.....	47
4.3	ESCOLHA DAS TÉCNICAS DE <i>IN-CONTEXT LEARNING</i>	50
4.4	SOFTWARE PARA INTERAGIR COM OS MODELOS VIA API.....	50
4.5	EXTRAÇÃO E TRATAMENTO DAS RESPOSTAS.....	51
4.6	ANÁLISE DOS RESULTADOS.....	51

5	RESULTADOS E DISCUSSÃO.....	53
5.1	RESULTADOS GERAIS.....	53
5.2	RESULTADOS POR DISCIPLINA.....	56
5.3	RESULTADOS POR ANO.....	59
5.4	RESULTADOS POR INSTITUTO.....	61
5.5	RESULTADOS EM RELAÇÃO A ACERTOS E ERROS.....	62
6	CONSIDERAÇÕES FINAIS.....	68
	REFERÊNCIAS.....	70

1 INTRODUÇÃO

A língua é uma ferramenta poderosa que desempenha um papel fundamental na comunicação e no desenvolvimento da sociedade. Através dela, transmitimos informações, expressamos ideias e estabelecemos interações significativas. Modelos de linguagem natural vêm sendo investigados há décadas na área conhecida como Processamento de Linguagem Natural (PLN). Com os recentes avanços tecnológicos, a maior disponibilidade de dados textuais digitalizados e o surgimento de métodos computacionais mais sofisticados, o PLN sofreu uma revolução, com destaque principal para advento dos Grandes Modelos de Linguagem (*Large Language Models*, LLMs) (DEVLIN *et al.*, 2019; BROWN *et al.*, 2020).

Os LLMs são modelos de *Deep Learning* capazes de descobrir padrões linguísticos a partir de grandes quantidades de textos de exemplo e gerar respostas coerentes. Estes padrões identificam distribuições de probabilidades para sequências de palavras que, por sua vez, podem ser empregadas para gerar textos sintéticos (MARS, 2022). Esses modelos têm demonstrado um potencial notável em uma variedade de aplicações, incluindo tradução automática, sumarização e geração de texto e resolução de questões (CHOWDHARY, 2020).

Em particular, no contexto educacional, as aplicações de LLMs vêm sendo investigadas por pesquisadores em diferentes desafios. Abordagens tanto da perspectiva de alunos quanto de professores são exploradas para dar escala a diversas atividades relacionadas à tutoria como a geração de conteúdo, melhora no engajamento e interação e a personalização das experiências de aprendizado (KASNECI *et al.*, 2023). A resolução automática de questões vem ganhando particular atenção nesse cenário. LLMs vem sendo testados em exames de ordem de advogados, provas de vestibular e em concursos públicos a fim de investigar os limites destes modelos. Esses experimentos vêm produzindo evidências de que LLMs são capazes de responder, com qualidade satisfatória, a questões complexas e escritas em inglês, inclusive em alguns casos melhor do que a média alcançada por seres humanos (OPENAI, 2023).

No Brasil, temos as instituições de ensino superior (IES) militares, algumas das quais apresentam forte ligação com a academia, com a oferta de cursos de graduação, mestrado e doutorado, como o Instituto Militar de Engenharia (IME) e o Instituto Tecnológica da Aeronáutica (ITA), ambos referência na formação de engenheiros e no desenvolvimento de tecnologia. Por conta do renome, da possibilidade de engajamento na carreira militar e das altas

relações entre número de candidatos e quantidade de vagas ofertadas, esses institutos possuem alguns dos vestibulares considerados mais difíceis do país (ARAÚJO; MATOS, 2021).

Este padrão de prova ainda é pouco explorado pela literatura, pois trata-se de conteúdo predominantemente de Ciências Exatas (Matemática, Física, Química) e escritos em português, o que leva a necessidade de habilidades como forte raciocínio lógico-matemático e entendimento do idioma. Um outro aspecto relevante é que soluções para estas questões não existem de forma numerosa na Internet, trazendo um maior potencial de dificuldade para os LLMs. Além disso, investigar a utilidade dos modelos para esses problemas e a confiabilidade dos resultados obtidos é crucial para o emprego adequado no contexto educacional.

1.1 PROBLEMA

Os Grandes Modelos de Linguagem são úteis na resolução de questões de vestibular complexas escritas na língua portuguesa?

1.2 JUSTIFICATIVA

Esta dissertação justifica-se pela investigação do potencial dos Grandes Modelos de Linguagem na resolução de questões complexas, identificando seus limites e avanços, especialmente no contexto brasileiro. Ao explorar a capacidade desses modelos em lidar com desafios significativos nos problemas propostos e ao avaliar seus resultados, busca-se contribuir para o aprofundamento do entendimento sobre as capacidades reais dessas tecnologias. Os LLMs podem desempenhar um papel relevante nesse contexto, oferecendo assistência aos estudantes na revisão e no aprimoramento do conhecimento adquirido, caso consigam fornecer respostas precisas para as questões.

Esta pesquisa também se justifica pela disponibilização de um novo conjunto de dados para a comunidade de Processamento de Linguagem Natural. A coleta e a preparação dos dados de questões de vestibular em instituições militares brasileiras podem enriquecer e diversificar os *datasets* disponíveis para treinar e avaliar novos LLMs mais eficazes e adaptados ao contexto

brasileiro (ALMEIDA *et al.*, 2023). Esses modelos, por sua vez, podem ser úteis não apenas para auxiliar estudantes no processo de aprendizagem e preparação para vestibulares, mas também para aprimorar a eficiência de plataformas educacionais online, fornecendo respostas mais completas e confiáveis para as dúvidas dos alunos.

1.3 OBJETIVOS

O desenvolvimento dessa pesquisa se baseia nos seguintes objetivos:

1.3.1 Objetivo geral

Investigar o desempenho dos Grandes Modelos de Linguagem em conjunto com técnicas de *in-context learning* na resolução de questões de vestibular complexas escritas em português.

1.3.2 Objetivos específicos

- a) Coletar, organizar e disponibilizar para a comunidade científica um novo conjunto de dados com questões de vestibular do IME e do ITA, que poderão ser usadas para treinamento e avaliação dos modelos;
- b) Descrever e discutir os conceitos que embasam a pesquisa;
- c) Construir e disponibilizar para a comunidade o software com os *prompts* utilizados para interagir com os modelos via API e para tratar os resultados;
- d) Comparar diferentes combinações de modelos e técnicas de engenharia de *prompt* na resolução de questões de vestibular;
- e) Analisar os resultados alcançados.

1.4 ORGANIZAÇÃO DO TRABALHO

O restante desta dissertação está estruturado da seguinte maneira: O segundo capítulo contém o referencial teórico que serve como base para a pesquisa, trazendo e discutindo os conceitos de *Deep Learning* com foco em texto, mecanismo de atenção e arquitetura *Transformer*, paradigmas dos modelos de linguagem e *in-context learning*. No terceiro, são debatidos os trabalhos relacionados e similares a essa pesquisa em conjunto com seus resultados. No quarto, temos os procedimentos metodológicos que foram empregados para o desenvolvimento da pesquisa, englobando a coleta dos dados, processo de escolha dos modelos e das técnicas de engenharia de *prompt* e como se dará a análise dos resultados. No quinto, temos a análise e discussão dos resultados e, no sexto e último capítulo, são feitas considerações finais da pesquisa e são apontadas limitações e possíveis trabalhos futuros.

2 REFERENCIAL TEÓRICO

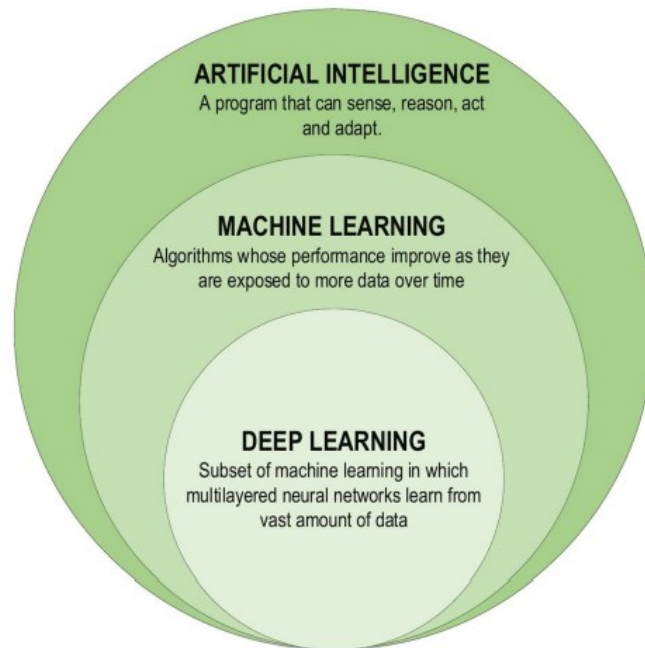
Visando o bom andamento dessa pesquisa, se faz necessária a discussão e a definição de alguns conceitos que servirão para clarear e orientar o seu entendimento. Este referencial teórico está dividido em quatro seções: a primeira discorre sobre as arquiteturas de *Deep Learning* aplicadas a problemas que envolvam texto; na segunda temos o aprofundamento da explicação sobre o mecanismo de atenção, arquitetura *Transformer* e os *Large Language Models* (LLMs) que revolucionaram o campo de PLN; a terceira é focada nos principais paradigmas do campo de PLN relacionados a modelos de Aprendizado de Máquina como pré-treinamento e *fine-tuning*; e a quarta e última seção discorre sobre *in-context learning* e as técnicas que buscam induzir os LLMs a seguirem comportamentos específicos no momento da inferência.

2.1 DEEP LEARNING APLICADO A PROBLEMAS DE PLN

Como ilustrado na figura 1 *Deep Learning* (DL) ou Aprendizado Profundo é uma subárea de *Machine Learning*, que por sua vez é uma subdivisão do campo de Inteligência Artificial (ALZUBAIDI *et al.*, 2021). Possui esse nome pois é projetado usando várias camadas de redes neurais artificiais, onde cada uma fornece uma interpretação diferente dos dados que são processados (ALZUBAIDI *et al.*, 2021).

Segundo Schmidhuber (2015) a ideia de redes neurais simples, com poucas camadas data de décadas ou até séculos atrás dependendo da definição. Com relação a *Deep Learning* os avanços mais concretos foram dados nas décadas de 60 e 70, mas acabaram encontrando dificuldades para avançar por volta dos anos 80 e acabou virando uma área de pesquisa com menos relevância por um tempo. Somente no começo do novo milênio que *Deep Learning* se tornou novamente relevante e a tecnologia começou a ser usada em larga escala, muito graças a combinação de três fatores: modelos com alto potencial, evolução do poder de computação e consequentemente barateamento de hardware e a disponibilidade de grandes conjuntos de dados anotados e abertos de forma livre para o público (SUN *et al.*, 2017).

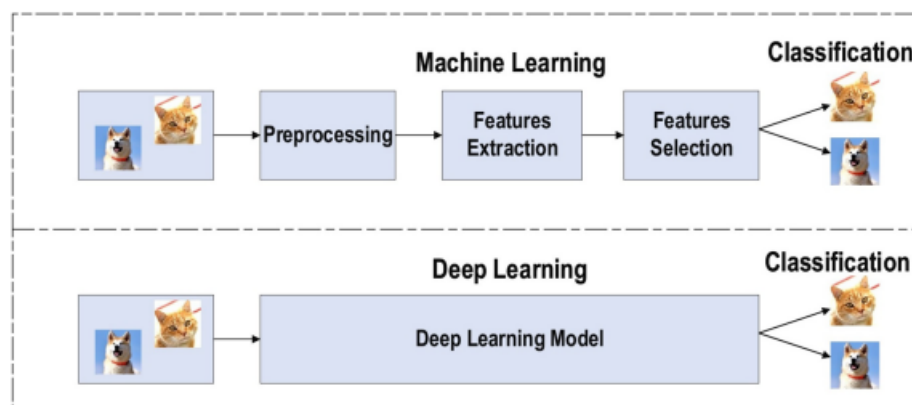
Figura 1 – Relações da área de *Deep Learning*



Fonte: Alzubaidi *et al.* (2021)

Alcançar a tarefa de classificação usando técnicas convencionais de aprendizado de máquina requer várias tarefas sequenciais, como pré-processamento dos dados, extração e seleção dos atributos, etapas que têm grande impacto no desempenho final do modelo. Uma das principais características da DL é que não requer essas atividades intermediárias que são projetadas por pessoas para funcionar, em vez disso usa uma grande quantidade de dados para mapear a entrada fornecida em relação aos rótulos informados (ALOM *et al.*, 2018).

Figura 2 – Comparação entre *Machine Learning* em relação a *Deep Learning*



Fonte: Alzubaidi *et al.* (2021)

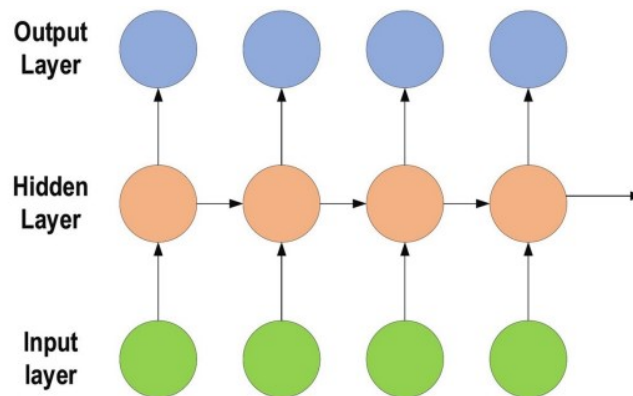
Quase todos os campos científicos sentiram o impacto dessa tecnologia e boa parte das indústrias e negócios foi transformada por meio do uso de DL. Atualmente, a performance de

modelos de Aprendizado Profundo em diversas áreas estão melhores do que o desempenho humano. Alguns exemplos são a previsão do tempo necessário para fazer entregas e a decisão sobre liberação de empréstimos (ALZUBAIDI *et al.*, 2021). Os vencedores do Prêmio Turing de 2019, considerado o Nobel da Computação, foram Yann LeCun, Geoffrey Hinton e Yoshua Bengio, três pioneiros no campo de *Deep Learning*.

Nessa seção focaremos nas principais arquiteturas de *Deep Learning* voltadas para aplicações de PLN, sendo elas: *Recurrent Neural Network* (RNN), *Long Short-Term Memory* (LSTM), *Gated Recurrent Unit* (GRU), *Bidirectional LSTM* (Bi-LSTM), *Sequence-to-Sequence* (Seq2Seq) e *Convolutional Neural Network* (CNN). *Attention* e *Transformer* serão tratados de forma separada posteriormente.

Recurrent Neural Network (RNN), que pode ser traduzido para Rede Neural Recorrente, é uma arquitetura de *Deep Learning* capaz de lidar especialmente com problemas em que a ordem dos dados é relevante como reconhecimento de fala, classificação de vídeos e processamento de textos (SHRESTHA; MAHMOOD, 2019).

Figura 3 – Exemplo da arquitetura RNN



Fonte: Alzubaidi *et al.* (2021)

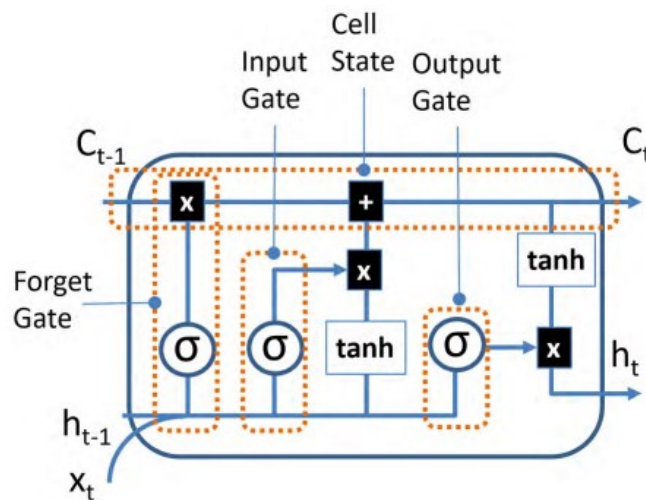
Possui esse nome porque a rede, como podemos ver na figura 3, processa uma sequência cada elemento por vez, mantendo em suas unidades ocultas um estado que contém informações sobre a história de todos os elementos anteriores da sequência. A qualquer instante um nó da rede recebe uma entrada, bem como valores ocultos que trazem informações das etapas anteriores (LECUN; BENGIO; HINTON, 2015).

Segundo LeCun, Bengio e Hinton (2015) existe um grande problema para se treinar modelos com RNN, pois as redes neurais geralmente precisam de *backpropagation* para

atualizar os seus pesos internos à medida que mais dados e épocas de treinamento são feitas. No entanto, as redes neurais recorrentes são muito sensíveis a isso. À medida que mais dados são inseridos ou vistos novamente durante o treinamento, os gradientes crescem rapidamente a ponto de “explodir” ou diminuem de forma veloz até desaparecer, fazendo com que a rede esqueça o que foi inserido nas primeiras sequências. Como uma alternativa para esse problema, foi proposto o LSTM.

Long Shot-Term Memory (LSTM) que foi proposta por Hochreiter e Schmidhuber (1997) é uma variação da implementação das RNNs que faz uso de camadas internas especiais com o objetivo de lembrar as entradas por um longo prazo.

Figura 4 – Bloco LSTM



Fonte: Shresta e Mahmood (2019)

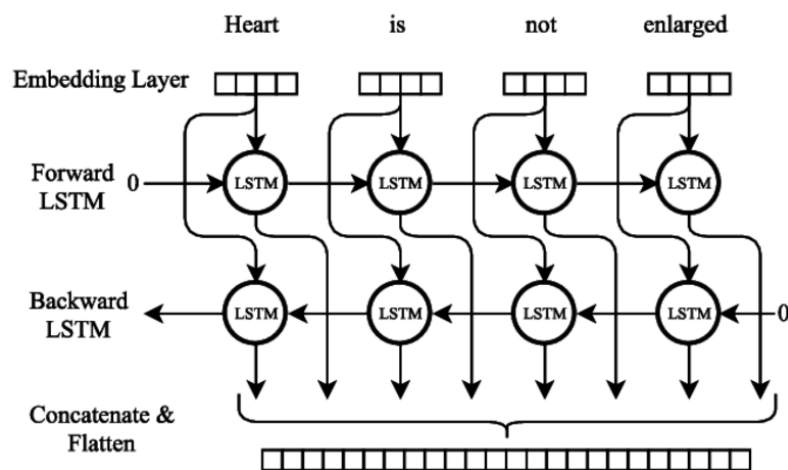
O treinamento da LSTM é realizado com o uso de gradiente descendente e como podemos observar pela figura 4 consiste em vários blocos de células de memória por onde flui um sinal que é regulado por três tipos de portões: entrada, saída e esquecimento que controlam o que é lido, escrito e armazenado na célula (SHRESTHA; MAHMOOD, 2019). A principal vantagem de usar essa abordagem é que o seu estado pode ser mantido e atualizado com computação e processamento mínimos, além de criar uma via segura para o processo de *backpropagation*, evitando assim o problema de desaparecimento e explosão do gradiente (LECUN; BENGIO; HINTON, 2015).

Existe uma alternativa conhecida como *Gated Recurrent Unit* (GRU), esse tipo de arquitetura só possui dois portões, o de *reset* e o de atualização, o que a torna menor e conseqüentemente mais fácil e rápido de se treinar. Tem a capacidade semelhante em algumas

tarefas em comparação com a LSTM, como modelagem musical e de fala (CHUNG *et al.*, 2014).

Já a Bi-LSTM (*Bidirectional Long Short-Term Memory*) é uma extensão da LSTM que é capaz de lidar com sequências de dados mantendo uma memória de longo prazo e aprendendo dependências de grande alcance.

Figura 5 – Representação da Bi-LSTM



Fonte: Cornegruta *et al.* (2016)

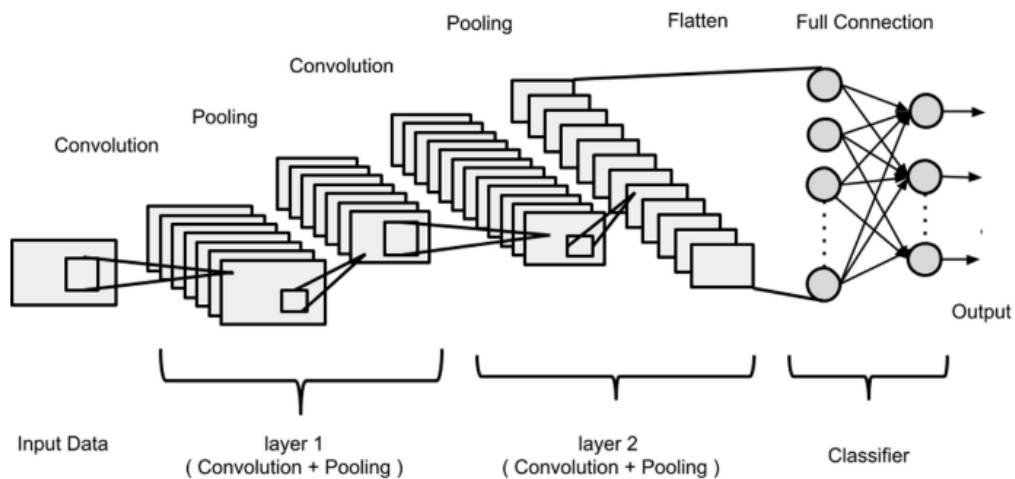
Segundo Schuster e Paliwal (1997) a principal característica da Bi-LSTM, como o nome diz, é a sua bidirecionalidade, onde a rede é dividida em duas camadas principais que se propagam em direções diferentes: uma para processar a sequência na ordem direta e outra na ordem inversa. Isso permite que a rede capture informações contextuais tanto das palavras anteriores quanto nas posteriores em uma sequência, o que melhora significativamente o desempenho em tarefas de processamento sequencial, como reconhecimento automático de fala e análise de sentimento (SCHUSTER; PALIWAL, 1997).

Adicionalmente temos a *Sequence-to-Sequence* (Seq2Seq) que é uma arquitetura de rede neural recorrente usada para resolver problemas de mapeamento de sequência, onde uma sequência de entrada é mapeada para uma sequência de saída (SUTSKEVER; VINYALS; LE, 2014). Essa abordagem é particularmente útil em tarefas como tradução automática, sumarização de textos e conversão de voz em texto. O modelo Seq2Seq é composto por duas LSTM que podem ter várias camadas: uma codificadora e outra decodificadora. A codificadora processa a sequência de entrada, como um texto em português, em um vetor de dimensão

reduzida, chamado de estado oculto. Esse estado é então passado para a camada decodificadora, que usa essa informação para gerar a sequência de saída, como o texto traduzido em francês.

Para finalizar temos a *Convolutional Neural Network* (CNN), também conhecida como Rede Neural Convolutiva, que de acordo com Patel e Patel (2021) surgiu como uma das melhores abordagens para resolver problemas do campo de Visão Computacional (VC) como classificação de imagens, reconhecimento facial e detecção de objetos, mas que também demonstrou posteriormente ser promissora em tarefas de Processamento de Linguagem Natural como a análise de sentimento.

Figura 6 – Arquitetura genérica de CNN



Fonte: Sengupta *et al.* (2020)

A figura 6 contém uma arquitetura comum de CNN que possui várias camadas de convolução e agrupamento (*pooling*) e uma camada totalmente conectada no final para gerar as classificações. De acordo com Sengupta *et al.* (2020), nesse exemplo as camadas de convolução extraem as características importantes da imagem de entrada considerando a posição dos pixels, enquanto as camadas de agrupamento reduzem a dimensionalidade, mas buscam manter as informações adquiridas. Uma imagem tem várias características como bordas, contornos, traços, texturas, orientação e cor, a CNN divide então uma imagem em termos dessas propriedades simples e as aprende como representações ao longo das diferentes etapas (SENGUPTA *et al.*, 2020).

Como já mencionado, pode-se usar CNN para problemas de PLN, obtendo como vantagem a sua capacidade de identificar padrões importantes em sequências de texto, capturando informações significativas em níveis diferentes de granularidade. A principal

diferença em relação ao uso com imagens é que no caso de texto as palavras precisam passar por uma etapa de processamento para serem transformadas em números, podendo então ser processadas pelas camadas internas da rede. Um dos problemas da CNN em texto é a dificuldade em preservar a ordem sequencial. Por outro lado, esta arquitetura requer um tempo de treinamento mais curto em comparação com outras abordagens de Aprendizado Profundo, como a RNN (PATEL; PATEL, 2021).

2.2 ATENÇÃO, ARQUITETURA *TRANSFORMER* E OS *LLM*

O conceito de atenção foi inserido no campo de Aprendizado Profundo por Mnih *et al.* (2014) e tinha o propósito de ser usado para problemas de Visão Computacional. Os autores comentam que embora os modelos de *Deep Learning* possuam desempenho superior no campo de VC eles demandam um alto custo computacional tanto para treinar quanto para fazer inferência. Uma Rede Neural Convolutiva profunda pode precisar de dias para ser treinada mesmo que as imagens passem por processo de redimensionamento e várias placas de vídeo (GPUs) sejam usadas em paralelo, além de demandarem segundos para realizar a inferência de novas instâncias. Isso se deve a natureza da CNN que faz o mapeamento dos filtros convolutivos em toda a imagem de entrada e a sua complexidade computacional é ao menos linear em relação ao número de pixels existentes (MNIH *et al.*, 2014).

A partir da literatura de Neurociência e Ciência Cognitiva, Mnih *et al.* (2014) comentam que uma propriedade importante da percepção humana é que ela não processa toda uma cena completa por vez, mas sim foca a atenção seletivamente em partes do campo visual para adquirir as informações quando e onde necessárias. Essas informações são posteriormente combinadas para construir uma representação interna da cena, que guia movimentos futuros dos olhos e nos auxilia no processo de tomada de decisão. Voltando ao campo de *Deep Learning*, a ideia foi desenvolver um *framework* para processamento visual baseado em atenção. Nele os recursos computacionais são concentrados em partes da cena e é possível economizar largura de banda pois menos pixels precisam ser processados, além de reduzir substancialmente a complexidade da tarefa (MNIH *et al.*, 2014).

Já no campo de Processamento de Linguagem Natural o mecanismo de atenção teve seus primeiros passos com Bahdanau, Cho e Bengio (2015) na área de tradução automática. Os

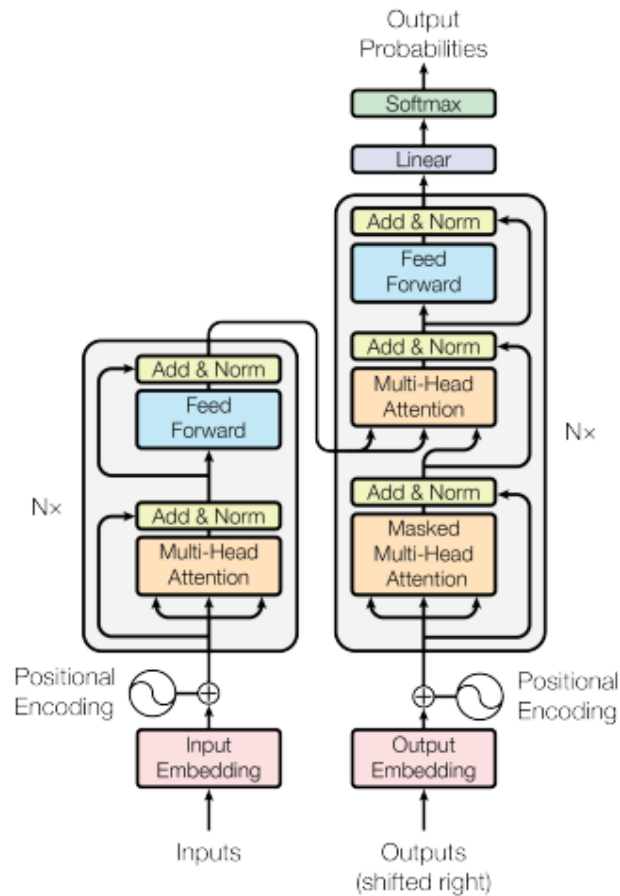
autores informam que a abordagem padrão da época era utilizar redes neurais que possuíam duas partes principais: codificação-decodificação, como a Seq2Seq. A primeira visava codificar a sentença de entrada de uma determinada língua em um vetor de tamanho fixo enquanto a camada decodificadora recebia essas informações e gerava a tradução em um segundo idioma. Segundo Bahdanau, Cho e Bengio (2015) um limitador dessa abordagem era justamente a geração desses vetores de tamanho fixo e então propuseram o uso da atenção para permitir que o modelo buscasse automaticamente por partes da sentença que eram mais relevantes para se prever uma palavra alvo. Com essa abordagem eles atingiram uma performance comparada ao estado da arte na época da tarefa de tradução do Inglês para o Francês.

Já em 2017 a área de PLN foi impactada pela publicação do artigo "*Attention is All You Need*". Vaswani *et al.* (2017) primeiro citam os avanços que foram possíveis com o uso do mecanismo de atenção, especialmente em problemas de modelagem sequencial, em que a ordem dos dados importa. Depois os autores comentam que na grande maioria dos casos a atenção é usada em conjunto com redes neurais recorrentes, o que não seria sempre necessário. Eles então propõem a arquitetura *Transformer*, que descarta a recorrência e utiliza inteiramente a atenção para focar diretamente nas posições que melhor descrevem as relações globais entre as entradas e as saídas (VASWANI *et al.*, 2017).

Os *Transformers* permitem que a rede neural aprenda relações complexas entre palavras ou tokens, entendendo o contexto em que elas estão para produzir representações das sentenças e documentos (VASWANI *et al.*, 2017). Junto com a ideia de modelos pré-treinados, a arquitetura *Transformer* se tornou ainda mais relevante pois com ela é possível realizar o treinamento de modelos em enormes conjuntos de dados genéricos de texto, usualmente extraídos de livros ou páginas da internet, que depois podem ser adaptados para tarefas específicas visando ganhos de performance (WOLF *et al.*, 2020).

Transformers foram originalmente propostos para uso em tradução automática, mas desde então são também usados em outras tarefas de Processamento de Linguagem Natural como classificação (YANG, 2019) e sumarização (LEWIS *et al.*, 2019).

Outro aspecto interessante dos *Transformers* é a sua eficiência computacional, que escala bem em relação aos dados de treinamento e ao tamanho da rede, além de facilitar o paralelismo, o que os torna adequados para o treinamento em grandes *datasets*. Isso a tornou a arquitetura de rede neural dominante para PLN nos últimos anos (WOLF *et al.*, 2020).

Figura 7 – Arquitetura do modelo *Transformer*

Fonte: Vaswani *et al.* (2017)

Na figura 7 podemos ver uma representação simplificada da arquitetura proposta por Vaswani *et al.* (2017) que consiste no *Transformer*. Em resumo esse é um modelo do tipo codificador-decodificador, em que a primeira parte, localizada no lado esquerdo da imagem, recebe uma sequência de entrada e a transforma em vetores, que são passados para o decodificador, localizado no lado direito da imagem, que gera uma sequência de saída (VASWANI *et al.*, 2017).

Entrando um pouco mais no detalhe o codificador e o decodificador possuem uma camada de atenção e outra *feed-forward*, que é uma arquitetura de rede neural simples, com camadas conectadas de entrada, ocultas e saída, onde os dados são processados em uma única direção. No decodificador temos ainda uma camada de atenção adicional que mapeia os tokens relevantes vindos do codificador (SINGH; MAHMOOD, 2021).

Segundo Singh e Mahmood (2021) a “atenção” promove a consulta da palavra de entrada em várias posições para determinar a relevância da que é processada no momento. Isso

é realizado para todas as palavras de entrada, o que ajuda a alcançar uma melhor representação contextual e gera vetores de atenção. Esses vetores são então alimentados para a camada *feed-forward*, onde são transformados em algo mais digerível para o próximo bloco. Em cada camada do *Transformer*, existem conexões residuais seguidas por uma etapa de normalização para acelerar o treinamento.

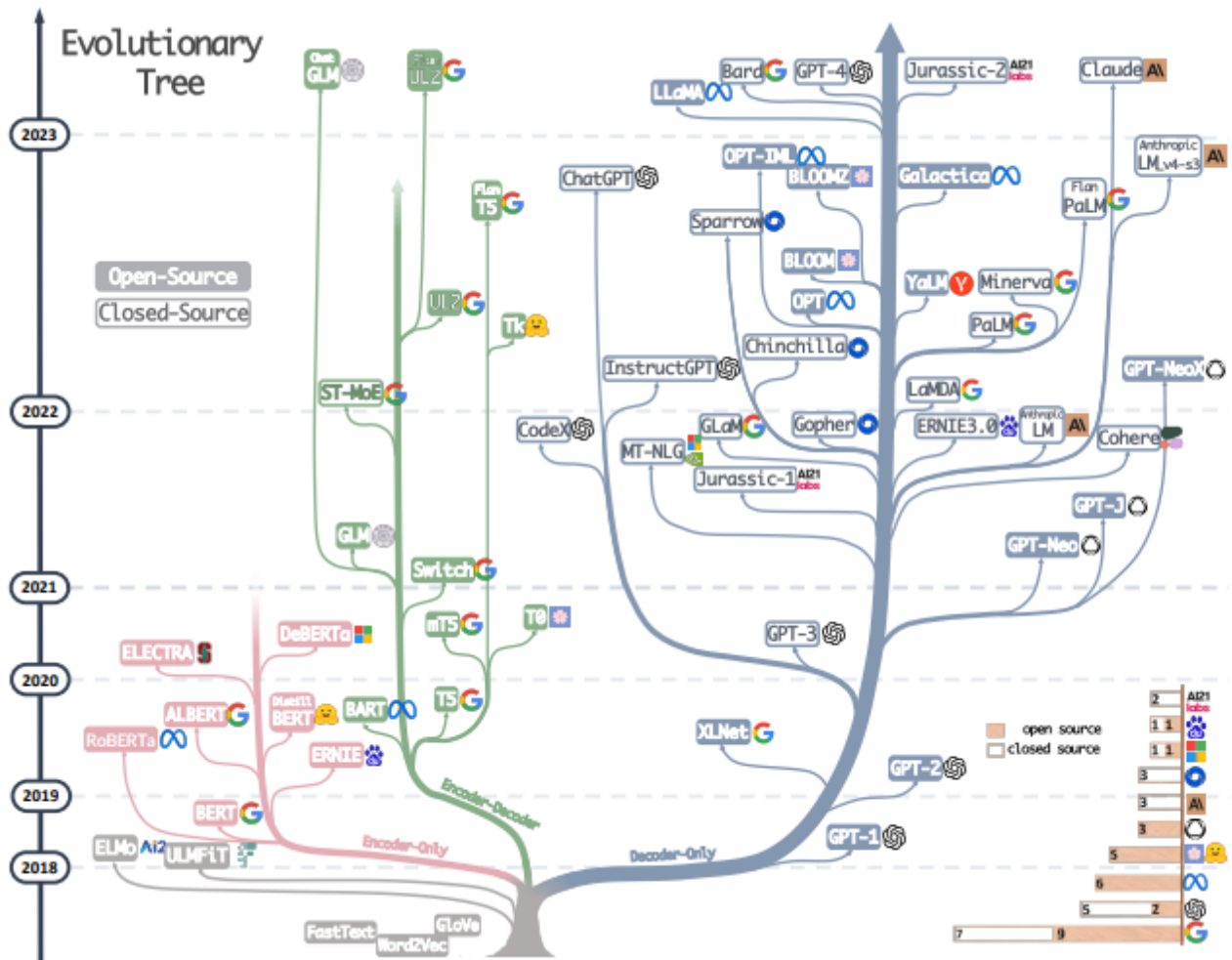
O decodificador é alimentado com a saída do codificador e a incorporação da sua própria entrada, e realiza o cálculo de atenção entre os dois. Isso determina a relevância dos tokens de entrada do *Transformer* em relação aos tokens de destino, pois o decodificador estabelece a representação vetorial entre os dois. O decodificador prevê a próxima palavra em várias etapas de tempo até que o token do final da frase seja gerado (SINGH; MAHMOOD, 2021).

Vale mencionar que o modelo *Transformer* completo proposto Vaswani *et al.* (2017) possui seis camadas codificadoras e outras seis decodificadoras e atingiu o estado da arte na tarefa de tradução após ser treinado por 12 horas em 8 GPUs.

Large Language Models (LLMs), também chamados de Grandes Modelos de Linguagem, são em linhas gerais modelos de *Deep learning* que utilizam a arquitetura *Transformer*, possuem número expressivo de parâmetros e são treinados de forma auto-supervisionada em grandes volumes de dados. Segundo Bowman (2023) esses modelos e os produtos criados em cima deles como o ChatGPT chamaram recentemente uma enorme atenção de empresas, jornalistas, formuladores de políticas públicas e acadêmicos de diversas áreas.

De acordo com Amatriain (2023) podemos separar os LLMs em duas classes pensando no seu uso: *foundation models* e *fine-tuned models*. Os primeiros, como o BERT (DEVLIN *et al.*, 2019) são definidos como qualquer modelo que é treinado em um grande volume de dados de forma auto-supervisionada, em escala e pode ser adaptado para uma variedade de tarefas posteriores. Quando esse modelo é treinado novamente em um volume reduzido de dados voltado para uma tarefa específica é chamado de *fine-tuned model*, pois foi afinado e otimizado nas especificidades de um determinado problema.

Figura 8 – Árvore evolutiva dos Grandes Modelos de Linguagem



Fonte: Yang *et al.* (2023)

Na figura 8 temos a árvore evolutiva dos Grandes Modelos de Linguagem feita por Yang *et al.* (2023) com os principais LLMs lançados, as organizações responsáveis e a informação se é um modelo fechado ou aberto para o público. Podemos citar duas características marcantes dos LLMs além do uso de *Transformers*: as leis de escala e as habilidades emergentes.

Sobre as leis de escala existem estudos que mostram que conforme a escala aumenta a performance dos modelos também aumenta. Zhao *et al.* (2023) dizem que visando estabelecer uma abordagem quantitativa para caracterizar o efeito de escala Kaplan *et al.* (2020) propuseram três fórmulas:

Figura 9 – Fórmulas da Lei de escala KM

$$\begin{aligned}
 L(N) &= \left(\frac{N_c}{N}\right)^{\alpha_N}, \quad \alpha_N \sim 0.076, N_c \sim 8.8 \times 10^{13} \\
 L(D) &= \left(\frac{D_c}{D}\right)^{\alpha_D}, \quad \alpha_D \sim 0.095, D_c \sim 5.4 \times 10^{13} \\
 L(C) &= \left(\frac{C_c}{C}\right)^{\alpha_C}, \quad \alpha_C \sim 0.050, C_c \sim 3.1 \times 10^8
 \end{aligned}$$

Fonte: Zhao *et al.* (2023)

Nas fórmulas [N] significa o tamanho do modelo em relação aos parâmetros, [D] o tamanho do *dataset*, [C] o poder computacional disponível para treinamento, [c] um valor determinado de orçamento para ser usado nesse processo e [L(.)] a performance levando em conta o *loss*, também conhecido como a taxa de aprendizado do modelo (ZHAO *et al.*, 2023). Kaplan *et al.* (2020) explicam que a primeira fórmula é voltada para o caso de modelos com um número limitado de parâmetros, treinados para convergir em conjuntos de dados suficientemente grandes, a segunda é para grandes modelos treinados em *datasets* limitados com a função de *early stopping* ativa, ou seja, o treinamento é encerrado se o modelo ficar algumas épocas sem melhorar, e por último temos o caso do treinamento feito com computação limitada, grande conjunto de dados e modelo de tamanho otimizado. A ideia central por trás desses cálculos é buscar prever de forma confiável o desempenho futuro de um LLM antes ou durante seu treinamento (ZHAO *et al.*, 2023).

Habilidades emergentes podem ser entendidas como capacidades que não estão presentes em modelos com poucos parâmetros, mas que aparecem a partir do ponto em que modelos que atingem determinada escala de número de parâmetros são treinados (WEI *et al.*, 2022b). Uma outra definição é que são habilidades que aparecem de forma inesperada aos quais alguns LLMs com milhões de parâmetros possuem, mas que não foram treinados diretamente visando essas tarefas (WEI *et al.*, 2022b). Algumas habilidades emergentes são a capacidade de dedução lógica (YANG *et al.*, 2023), programação, aritmética (BOWMAN, 2023) e o *in-context learning* (ZHAO *et al.*, 2023) que será tratado na seção 2.4.

Para finalizar, Kaddour *et al.* (2023) citam alguns desafios presentes em se lidar com LLMs como o alto custo da etapa de pré-treinamento, que pode chegar a exigir milhares de horas computacionais que por sua vez custam milhões de dólares e consomem quantidades de energia equivalentes às usadas por várias famílias dos EUA anualmente. Como outros pontos

temos as alucinações, que são textos gerados por esses modelos que são fluentes e naturais, mas apresentam dados incorretos e a questão da base de informações que fica parada no tempo ao fim do treinamento e com isso vai ficando desatualizada.

2.3 OS PARADIGMAS DOS MODELOS DE LINGUAGEM

De acordo com Liu *et al.* (2023) ocorreram duas grandes mudanças recentes no campo de Processamento de Linguagem Natural em relação aos modelos de Aprendizado de Máquina. Os autores dizem que anteriormente, até o fim da década passada, o aprendizado supervisionado era dominante. Primeiramente com algoritmos tradicionais que dependem bastante do processo de engenharia de atributos para obter desempenho satisfatório e depois com o uso mais frequente de *Deep Learning*, onde o trabalho foi deslocado da modelagem dos dados para a construção ótima de blocos de rede neural que melhor conseguissem mapear as entradas com as saídas desejadas.

Já a partir de 2017 tivemos a primeira grande alteração no campo com o surgimento do paradigma “pré-treinamento e *fine-tuning*”. Nesse cenário um artefato é treinado a partir de uma arquitetura de rede neural fixa em enormes conjuntos de dados de forma auto-supervisionada, ou seja, sem as variáveis alvo e possuem o objetivo de entender a probabilidade dos dados textuais observados. O modelo de linguagem pré-treinado é então adaptado para diferentes tarefas posteriormente, introduzindo parâmetros adicionais e dados em escala bem menor, otimizando-o para os detalhes da tarefa que se deseja resolver. Dentro deste paradigma, o foco voltou-se principalmente para a engenharia de detalhes técnicos como novas funções que poderiam ser utilizadas durante o treinamento para que o afinamento ficasse melhor ou mais rápido (LIU *et al.*, 2023).

A segunda grande mudança, segundo Liu *et al.* (2023), aconteceu a partir de 2021, nela fomos de “pré-treinamento e *fine-tuning*” para o surgimento do paradigma de “pré-treinamento, *prompt* e predição”. Nesse caso não é necessário o investimento para se construir um novo *dataset* supervisionado de tamanho razoável e nem em hardware para continuar o treinamento do modelo. A grande vantagem desse método é que, dado um conjunto de *prompts* apropriados, um único modelo pode ser usado para resolver várias tarefas diferentes. Outro ponto relevante é que dessa forma é preenchida a lacuna entre os objetivos do pré-treinamento e da atividade

posterior, permitindo uma melhor utilização das informações adquiridas durante a primeira etapa. No entanto, há um problema, esse método introduz a necessidade de busca da melhor forma de interagir com o LLM para solucionar uma atividade, área denominada de engenharia de *prompt*.

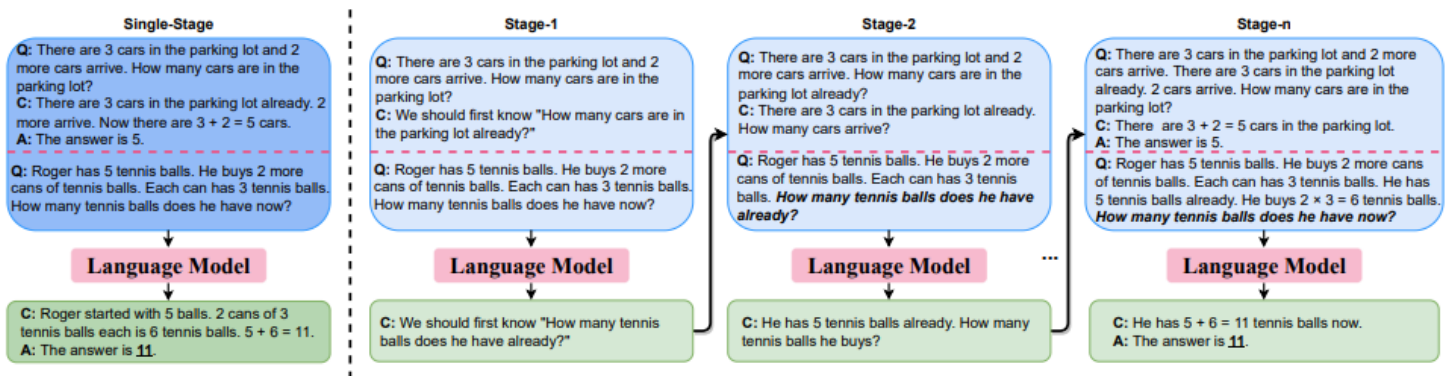
Importante ressaltar que embora tenhamos essa diferença na relevância das técnicas ao longo dos anos como apontado, um paradigma não substituiu o outro em sua totalidade. Hoje temos diversos tipos de modelos que recaem em cada uma dessas categorizações separadamente e os que fazem a combinação desses paradigmas. Por exemplo temos os LLMs pré-treinados que podem ser usados diretamente para uma tarefa ou podem passar por etapas de *fine-tuning* para apresentarem melhor aspecto de conversação com os usuários (OPENAI, 2023), ficarem especializados em áreas do conhecimento como Medicina (SINGHAL *et al.*, 2023; WU *et al.*, 2023) ou em tarefas específicas como a geração de código (LUO *et al.*, 2023; GUNASEKAR *et al.*, 2023). Nesta pesquisa o foco será no paradigma “pré-treinamento, *prompt* e predição” que será mais explorada a seguir.

2.4 IN-CONTEXT LEARNING

Gu *et al.* (2023) definem *prompt* como informações adicionais ou dicas que são submetidas a um modelo para guiar o seu comportamento ou ajudar a executar uma tarefa específica. Já *in-context learning* é um método que provê instruções ou demonstrações de como uma tarefa deve ser realizada aos Grandes Modelos de Linguagem visando a resolução de novos casos sem que o modelo precise passar por treinamento adicional (GU *et al.*, 2023). Outros autores como Qiao *et al.* (2023) usam o termo engenharia de *prompt* como sinônimo para *in-context learning*.

Qiao *et al.* (2023) subdividem as técnicas de engenharia de *prompt* em dois grandes grupos, as de única etapa e as de múltiplas etapas. Kaddour *et al.* (2023) por sua vez as denomina de único turno e múltiplos turnos, porém, com a mesma ideia.

Figura 10 – Funcionamento dos *prompts* de um e múltiplos estágios



Fonte: Qiao *et al.* (2023)

Como é possível observar na figura 10, temos na esquerda o caso de único estágio onde um exemplo resolvido é passado junto de um novo caso a ser predito e a ideia é que o modelo traga a resposta de imediato, encerrando assim a interação. Já no exemplo de múltiplos estágios busca-se explicitamente inserir questões adicionais junto com a extração e uso de respostas intermediárias visando reduzir uma possível lacuna do processamento dos LLMs (QIAO, 2023).

Nesta seção teremos o aprofundamento das técnicas de *in-context learning* que foram usadas neste trabalho, todas elas consideradas de único turno: *Zero-Shot*, *Few-Shot*, *Chain-of-Thought*, *Zero-shot Chain-of-Thought* e *Plan-and-Solve*.

2.4.1 Zero-shot e Few-shot

No artigo de lançamento do GPT-3, Brown *et al.* (2020) trazem a diferença entre o aprendizado humano, que necessita de alguns poucos exemplos para conseguir realizar a maioria das tarefas relacionadas a linguagem em relação aos Grandes Modelos de Linguagem que precisam ingerir uma quantidade enorme de textos durante a etapa de pré-treinamento.

Também é argumentado no trabalho que o aumento de escala dos LLM, no caso do GPT-3 para 175 bilhões de parâmetros, 10 vezes maior que os modelos de linguagem divulgados antes, aliado a técnicas de *in-context learning*, possibilitaram aumentar substancialmente a performance em *benchmarks*, em alguns casos sendo competitivo até com modelos que

passaram pela etapa de *fine-tuning* e que eram considerados estado da arte em determinadas tarefas (BROWN *et al.*, 2020).

Segundo Brown *et al.* (2020) o framework de aprendizado supervisionado era o mais usado anteriormente para se atingir desempenhos elevados em determinadas tarefas do campo de Processamento de Linguagem Natural. Porém, geralmente necessita-se de um considerável conjunto de dados anotados com qualidade. Para produzir um dataset de qualidade é necessário investimento para empregar pessoas especializadas no problema que se quer resolver, além do tempo de dedicação para operacionalização e revisão dos conjuntos de dados disponíveis. Ainda temos ao fim o custo de hardware para treinar um modelo do zero, ou no caso de um *Large Language Models*, usar um grande modelo pré-treinado e prosseguir com a etapa de *fine-tuning*, de forma a otimizar o modelo para uma tarefa específica.

Figura 11 – Técnicas das técnicas *Zero-shot*, *One-shot* e *Few-shot*

The three settings we explore for in-context learning

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 cheese => ..... ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← example
3 cheese => ..... ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← examples
3 peppermint => menthe poivrée ←
4 plush girafe => girafe peluche ←
5 cheese => ..... ← prompt
```

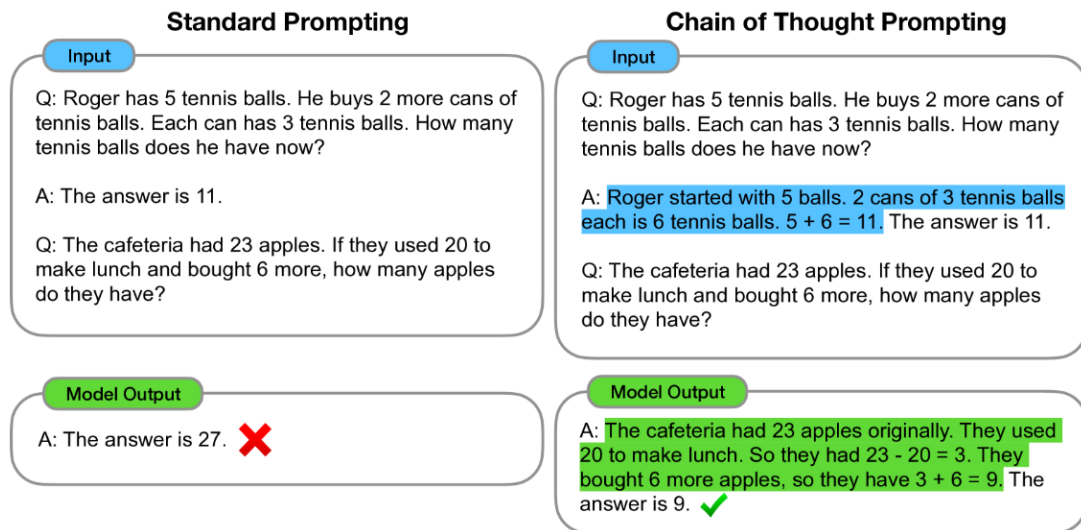
Fonte: Brown *et al.* (2020)

De acordo com a imagem é possível vermos as diferenças entre as técnicas apresentadas. *Zero-shot* é a mais simples e é geralmente usada como ponto de partida para comparações com outras abordagens mais avançadas. Nessa abordagem é passada a descrição da tarefa junto com a instância que deve ser feita a inferência. Na sequência temos as técnicas de *one-shot* e *few-shot* que possuem ideia similar, além da instrução da tarefa, também são enviados exemplos que visam ensinar ao modelo a forma correta de concluir o trabalho. A principal diferença entre elas é a quantidade de exemplos enviados, enquanto na *one-shot* é passado uma instância, no caso do *few-shot* são enviados um número arbitrário de eventos maior que do um (BROWN *et al.*, 2020).

De acordo com Brown *et al.* (2020), GPT-3 utilizando a técnica de *one-shot* empatou com o modelo RAG, considerado até então o estado da arte no *dataset* TriviaQA, de perguntas e respostas, em 68.8 de acurácia. Já com o uso de *few-shot* atingiu o novo estado na arte com 71.2. Ao contrário do GPT-3, o RAG passou por *fine-tuning* para essa tarefa específica. Outro caso interessante foi o LAMBADA, conjunto de dados que avalia a capacidade dos modelos de entender uma passagem e prever a última palavra da sentença. *Zero*, *one* e *few-shot* tiveram performance melhor do que o modelo estado da arte até então (BROWN *et al.*, 2020).

2.4.2 Chain-of-Thought

Wei *et al.* (2022a) argumentam que somente aumentar o tamanho de parâmetros dos LLMs não provou ser suficiente para alcançar performance elevada em tarefas mais desafiadoras como as de aritmética e conhecimentos gerais. Eles apresentam o método *Chain-of-Thought* (CoT) que visa destravar a habilidade de raciocínio baseado em duas ideias: a primeira é que técnicas que almejam atingir bom desempenho em problemas aritméticos podem se beneficiar da linguagem e a segunda é que como já mencionado os *Large Language Models* apresentam a capacidade de *in-context learning* diretamente via *prompt*.

Figura 12 – Diferença entre *One-shot* e *Chain-of-Thought*

Fonte: Wei *et al.* (2022a)

Chain-of-Thought, também conhecido como cadeia de pensamento, é definida por Wei *et al.* (2022a) como uma técnica que consiste em três partes indispensáveis como demonstrado na figura 12: input, a cadeia de pensamento ou passos para se chegar no resultado e a resposta. Esse método de engenharia de *prompt* visa adicionar uma série de etapas intermediárias de raciocínio em linguagem natural que levem o modelo a replicar esse comportamento em novos casos.

De acordo com Wei *et al.* (2022a), CoT possui quatro elementos que a torna atrativa para facilitar o raciocínio nos *Large Language Models*:

- Permite que os modelos decomponham problemas complexos em etapas intermediárias, o que significa que computação pode ser alocada melhor nas partes mais difíceis;
- Uma cadeia de pensamento fornece uma janela interpretável para o comportamento do modelo, sugerindo como ele pode ter chegado a uma resposta específica e fornecendo oportunidades para depurar onde o caminho do raciocínio deu errado;
- O raciocínio com cadeia de pensamento pode ser usado para tarefas como problemas matemáticos, raciocínio de senso comum e manipulação simbólica, e é potencialmente aplicável a qualquer tarefa que os humanos possam resolver por meio da linguagem.

- Por último, o raciocínio da cadeia de pensamento pode ser prontamente gerado em modelos de linguagem disponíveis no mercado suficientemente grandes, simplesmente incluindo exemplos de sequências de cadeia de pensamento no *prompt*.

A técnica de CoT é construída em conjunto com a de *one-shot* ou *few-shot* a depender de quantos exemplos com os passos para se chegar na resposta são enviados. Com o uso de *Chain-of-Thought* em conjunto ao modelo PaLM, que possui 540 bilhões de parâmetros e foi desenvolvido pelo Google, foi possível atingir o novo estado da arte nos *datasets* de problemas aritméticos GSM8K, SVAMP e MAWPS. Os resultados atingidos foram regularmente melhores do que as métricas usando *prompt* padrão sem as etapas de raciocínio (WEI *et al.*, 2022a).

Wei *et al.* (2022a) elencam algumas limitações do método: embora a ideia seja simular o raciocínio humano, ela não responde à questão de se as redes neurais de um LLM estão realmente “pensando” no momento da inferência. Outro ponto é que não existe garantia de que etapas intermediárias de raciocínio garantam corretude, o que pode gerar respostas incorretas ou respostas corretas a partir de passos errados. Por último temos que a utilidade de *Chain-of-Thought* somente foi observada em modelos com bilhões de parâmetros, o que a torna custosa para ser usada em aplicações da indústria.

2.4.3 Zero-shot Chain-of-Thought

A técnica de *Zero-shot Chain-of-Thought* busca equilibrar os pontos positivos e negativos de *Zero-shot* e *Chain-of-Thought*. Enquanto na primeira temos o lado bom de instruir os modelos a realizar uma tarefa sem a necessidade de criarmos exemplos de qualidade, temos o lado ruim de que essa técnica geralmente não gera passos intermediários e desse modo atingem resultados menos expressivos. *Chain-of-Thought* é basicamente o contrário: enquanto tem a capacidade de gerar passos intermediários de raciocínio, o que geralmente melhora os resultados das predições, possui o lado negativo de necessitar de exemplos similares com qualidade para ser usada, exemplos esses que a depender do problema a ser resolvido precisam ser construídos por especialistas.

Figura 13 – Ilustração da técnica de *Zero-shot CoT*

<p style="text-align: center;">(a) Few-shot</p> <p>Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now? A: The answer is 11.</p> <p>Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there? A:</p> <hr/> <p>(Output) The answer is 8. X</p>	<p style="text-align: center;">(b) Few-shot-CoT</p> <p>Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now? A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.</p> <p>Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there? A:</p> <hr/> <p>(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are $16 / 2 = 8$ golf balls. Half of the golf balls are blue. So there are $8 / 2 = 4$ blue golf balls. The answer is 4. ✓</p>
<p style="text-align: center;">(c) Zero-shot</p> <p>Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there? A: The answer (arabic numerals) is</p> <hr/> <p>(Output) 8 X</p>	<p style="text-align: center;">(d) Zero-shot-CoT (Ours)</p> <p>Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there? A: Let's think step by step.</p> <hr/> <p>(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓</p>

Fonte: Kojima *et al.* (2022)

Como podemos ver na figura 13, a técnica *Zero-shot-CoT* além do enunciado da questão a ser resolvida também inclui uma instrução ao modelo para ser considerada no momento da inferência. No exemplo (d) temos o comando “Vamos pensar passo a passo” (KOJIMA *et al.*, 2022, p. 2, tradução nossa). A ideia é remover o trabalho de curadoria para construir ou selecionar exemplos de qualidade ao mesmo tempo em que sejam gerados passos intermediários, e que assim os resultados sejam melhores (KOJIMA *et al.*, 2022).

Kojima *et al.* (2022) afirmam que com essa técnica foi possível melhorar os resultados em comparação com o método de *zero-shot* em vários testes aritméticos, simbólicos e lógicos. Por exemplo nos *datasets* relacionados a Matemática MultiArith e GSM8K, a acurácia saltou de 17.7% para 78.7% e 10.4% para 40,7% respectivamente. Mesmo assim os resultados ficam aquém em comparação a *few-shot CoT*.

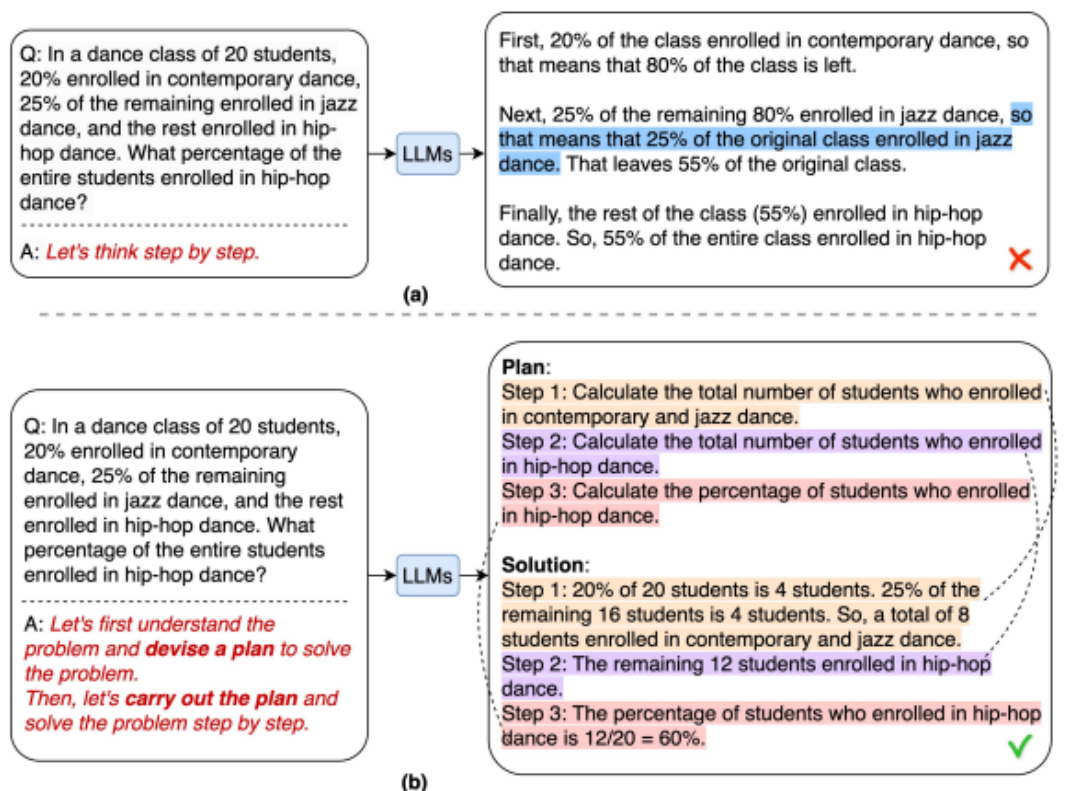
Como limitação ao trabalho, os autores comentam sobre a falta de detalhes sobre os dados usados para treinamento dos LLM. A falta de acesso aos dados levanta dúvidas sobre a memorização de certas informações por parte dos modelos, tornando-os enviesados com a possibilidade de saber a resposta de antemão para algumas perguntas de forma a contaminar a avaliação (KOJIMA *et al.*, 2022).

2.4.4 Plan-and-Solve

Wang *et al.* (2023) partem dos resultados expressivos atingidos com *Zero-shot Chain-of-Thought* mas comentam que ela sofre de 3 problemas principais. Em uma análise de 100 questões de Matemática selecionadas de forma aleatória do *dataset* GSM8K 46 foram respondidas de forma incorreta pelos seguintes problemas: erro de cálculo (7), etapas de raciocínio ignoradas (12) e problema com o entendimento da questão (27).

A partir daí Wang *et al.* (2023) apresentam o método *Plan-and-Solve* (PS), técnica também considerada de única etapa que permite aos LLMs elaborar explicitamente um plano para resolver um determinado problema e gerar o processo de raciocínio intermediário antes de prever a resposta final para a questão de entrada. Ao contrário das abordagens anteriores de CoT com *few-shot*, em que exemplos de demonstração com passo a passo são incluídos no *prompt*, o método PS *zero-shot* não requer exemplos de demonstração e seu *prompt* contém somente o problema em si e uma instrução simples.

Figura 14 – Comparação entre *Zero-shot CoT* e *Plan-and-Solve*



Fonte: Wang *et al.* (2023)

Existem duas variações do método *Plan-and-Solve*, o primeiro e que está na figura 14 consiste na instrução “Vamos primeiro entender o problema e elaborar um plano para resolvê-lo. Então, vamos executar o plano e resolver o problema passo a passo.” (WANG *et al.*, 2023, p. 7, tradução nossa). Já o segundo, apelidado de PS+, é um pouco diferente: “Vamos primeiro entender o problema, extrair variáveis relevantes e os números correspondentes e fazer um plano. Então, vamos executar o plano, calcular as variáveis intermediárias (atenção ao cálculo numérico correto e ao bom senso), resolver o problema passo a passo e responder.” (WANG *et al.*, 2023, p. 7, tradução nossa).

Com o uso da técnica PS+ foi possível melhorar a acurácia de *Zero-shot-CoT* em ao menos 5% em vários *datasets* de problemas aritméticos como MultiArith que subiu de 83.8% para 91.8%, o AddSub que subiu de 85.3% para 92.2%, o AQuA que subiu de 38.9% para 46%, o SingleEq que subiu de 88.1% para 94.7% e SVAMP que subiu de 69.9% para 75.7%. A única exceção foi o GSM8K em que o ganho foi de 2.9%. Em comparação com CoT, dos 6 testes, PS+ ficou pior do que CoT em 3, com uma média de acurácia ligeiramente pior no geral 76.6% contra 77.6%. Resultados similares também foram encontrados em problemas de conhecimentos gerais e manipulação simbólica como concatenação das últimas letras de um nome e o teste de cara ou coroa onde várias instruções sobre a virada ou não da moeda são passadas e a ideia é que o modelo consiga descobrir a sua posição final (WANG *et al.*, 2023).

São duas as limitações desse método: a primeira é que se faz necessário um certo esforço para elaborar e testar os *prompts* que melhor vão guiar os LLMs a gerar passos intermediários corretos e o segundo é que *Plan-and-Solve* é capaz de ajudar a evitar os erros relacionados a cálculos incorretos e passos intermediários ausentes, mas peca quando o problema é a falta de entendimento semântico da questão (WANG *et al.*, 2023).

3 TRABALHOS RELACIONADOS

Nessa seção discutiremos sobre trabalhos relacionados ou similares a essa pesquisa que foram encontrados após revisão da literatura.

A OpenAI (2023) disponibilizou no relatório técnico do GPT-4 uma demonstração de como os seus principais modelos desempenharam em diversos testes acadêmicos e profissionais aplicados nos Estados Unidos da América, assim como sua posição em comparação com as pessoas que realizaram essas provas.

Tabela 1 – Desempenho dos LLMs em diferentes testes aplicados nos EUA

Exam	GPT-4	GPT-4 (no vision)	GPT-3.5
Uniform Bar Exam (MBE+MEE+MPT)	298 / 400 (~90th)	298 / 400 (~90th)	213 / 400 (~10th)
LSAT	163 (~88th)	161 (~83rd)	149 (~40th)
SAT Evidence-Based Reading & Writing	710 / 800 (~93rd)	710 / 800 (~93rd)	670 / 800 (~87th)
SAT Math	700 / 800 (~89th)	690 / 800 (~89th)	590 / 800 (~70th)
Graduate Record Examination (GRE) Quantitative	163 / 170 (~80th)	157 / 170 (~62nd)	147 / 170 (~25th)
Graduate Record Examination (GRE) Verbal	169 / 170 (~99th)	165 / 170 (~96th)	154 / 170 (~63rd)
Graduate Record Examination (GRE) Writing	4 / 6 (~54th)	4 / 6 (~54th)	4 / 6 (~54th)
USABO Semifinal Exam 2020	87 / 150 (99th - 100th)	87 / 150 (99th - 100th)	43 / 150 (31st - 33rd)
USNCO Local Section Exam 2022	36 / 60	38 / 60	24 / 60
Medical Knowledge Self-Assessment Program	75 %	75 %	53 %
Codeforces Rating	392 (below 5th)	392 (below 5th)	260 (below 5th)
AP Art History	5 (86th - 100th)	5 (86th - 100th)	5 (86th - 100th)
AP Biology	5 (85th - 100th)	5 (85th - 100th)	4 (62nd - 85th)
AP Calculus BC	4 (43rd - 59th)	4 (43rd - 59th)	1 (0th - 7th)
AP Chemistry	4 (71st - 88th)	4 (71st - 88th)	2 (22nd - 46th)
AP English Language and Composition	2 (14th - 44th)	2 (14th - 44th)	2 (14th - 44th)
AP English Literature and Composition	2 (8th - 22nd)	2 (8th - 22nd)	2 (8th - 22nd)
AP Environmental Science	5 (91st - 100th)	5 (91st - 100th)	5 (91st - 100th)
AP Macroeconomics	5 (84th - 100th)	5 (84th - 100th)	2 (33rd - 48th)
AP Microeconomics	5 (82nd - 100th)	4 (60th - 82nd)	4 (60th - 82nd)
AP Physics 2	4 (66th - 84th)	4 (66th - 84th)	3 (30th - 66th)
AP Psychology	5 (83rd - 100th)	5 (83rd - 100th)	5 (83rd - 100th)
AP Statistics	5 (85th - 100th)	5 (85th - 100th)	3 (40th - 63rd)
AP US Government	5 (88th - 100th)	5 (88th - 100th)	4 (77th - 88th)
AP US History	5 (89th - 100th)	4 (74th - 89th)	4 (74th - 89th)
AP World History	4 (65th - 87th)	4 (65th - 87th)	4 (65th - 87th)
AMC 10 ³	30 / 150 (6th - 12th)	36 / 150 (10th - 19th)	36 / 150 (10th - 19th)
AMC 12 ³	60 / 150 (45th - 66th)	48 / 150 (19th - 40th)	30 / 150 (4th - 8th)
Introductory Sommelier (theory knowledge)	92 %	92 %	80 %
Certified Sommelier (theory knowledge)	86 %	86 %	58 %
Advanced Sommelier (theory knowledge)	77 %	77 %	46 %
Leetcode (easy)	31 / 41	31 / 41	12 / 41
Leetcode (medium)	21 / 80	21 / 80	8 / 80
Leetcode (hard)	3 / 45	3 / 45	0 / 45

Fonte: OpenAI (2023)

À primeira vista o GPT-4 alcançou um desempenho de destaque em comparação com participantes humanos, especialmente em testes como *Scholastic Assessment Test* (SAT), prova usada como processo de admissão em cursos de graduação em universidades americanas, *Law School Admission Test* (LSAT), prova usada para entrada em cursos de Direito e *Bar Exam*, prova que formandos de Direito precisam fazer para obter a licença e praticar a advocacia, similar ao exame da OAB. No entanto, em um *benchmark* bilíngue, com atividades em inglês e chinês, Zhong *et al.* (2023) descobriram que o GPT-4 pode ser amplamente influenciado por tarefas com muitos dados disponibilizados online ou que estejam em inglês, como o SAT. Nesses casos os modelos superam as médias humanas, alcançando quase os melhores desempenhos. Já para tarefas com menos dados, em outra língua ou que precisem de conhecimento específico, os modelos acabam se saindo pior, como evidenciado em seu desempenho no teste de qualificação de advogados e em questões de concursos públicos da China. Os resultados sugerem que o desempenho exitoso pode ser resultado de memorização e não devido ao aprendizado. Também foi descoberto que o GPT-4 tem melhor desempenho em disciplinas em que a leitura e a linguagem têm maior peso, como História e Geografia, enquanto seu desempenho cai em disciplinas que exigem forte habilidade de resolução de problemas, como Matemática, Física e Química (ZHONG *et al.*, 2023).

Zhang *et al.* (2023) criaram o *dataset Chinese College Entrance Exam* (GAOKAO) e analisaram o desempenho do ChatGPT. Esse exame, que serve tanto para avaliar o conhecimento dos concluintes do ensino médio e como porta de entrada para universidades chinesas, é considerado rigoroso e possui questões de Matemática, Chinês, Inglês, Física, Química, Biologia, Política, História e Geografia. Como esperado, as melhores pontuações foram nas questões de Inglês com uma taxa de acerto acima dos 70% e piores em questões de Matemática, Física e Química ou que necessitavam da leitura de longas passagens em chinês.

Arora, Singh e Mausam (2023) realizaram um experimento em que testaram diferentes *Large Language Models* em conjunto com algumas técnicas de engenharia de *prompt* para resolver questões do *Joint Entrance Examination* (JEE) – *Advanced*. Esta prova é considerada difícil e com baixa taxa de aprovação, aproximadamente 5%, e é utilizada para acesso aos Institutos Indianos de Tecnologia (IIT), que são referência no país.

Os autores coletaram ao todo 450 questões, em inglês e que não necessitavam das imagens do enunciado para serem resolvidas, de Física, Química e Matemática das últimas 7 edições do exame. Os modelos utilizados foram text-davinci-003 (GPT-3), gpt-3.5-turbo (GPT-

3.5) e gpt-4-0314 (GPT-4), todos da OpenAI, junto de duas técnicas de *in-context learning*: *Chain-of-Thought* e *Self-Consistency*. Foram empregadas análises quantitativa e qualitativa dos resultados.

Tabela 2 – Resultados do JEE por área do conhecimento

	Random	GPT-3	GPT-3.5	GPT-4	GPT-4+CoT	GPT-4+CoT+SC
Chemistry	0.106	0.119	0.241	0.428	0.493	0.477
Physics	0.096	0.104	0.189	0.334	0.329	0.294
Mathematics	0.103	0.107	0.157	0.231	0.281	0.224
Total	0.102	0.110	0.191	0.316	0.358	0.319

Fonte: Arora, Singh e Mausam (2023)

Na tabela 1 podemos ver a performance dos modelos e das técnicas utilizadas. O GPT-3 apresenta um desempenho similar ao aleatório, já o GPT-3.5 e o GPT-4, versões mais recentes, demonstram de forma consistente performance superior. O GPT-4 é muito superior ao GPT-3.5 por uma grande margem de 12,5 pontos, o aumento de desempenho é maior em Química, seguido de Física e, por último, Matemática. A melhor nota total se dá pela combinação do GPT-4 e CoT. Nesse caso os autores se referem na verdade a técnica de *Zero-shot Chain-of-Thought* de Kojima *et al.* (2022). Somente em Física que o GPT-4 sozinho se saiu melhor.

Já para a análise qualitativa foram reunidas 100 questões respondidas pela combinação do GPT-4+CoT. As respostas foram inspecionadas e posteriormente categorizadas. Os maiores destaques negativos foram erro de cálculo (31%), onde uma operação como elevar um número ao quadrado é feita algebricamente errada, seguido por erro conceitual (26%), onde o modelo é incapaz de recuperar os conceitos necessários para resolver o problema.

Arora, Singh e Mausam (2023) concluem comentando sobre a capacidade dos LLM, como o GPT-4, que é capaz de responder corretamente algumas questões de forma parecida com uma pessoa, o que é impressionante dado o nível de raciocínio necessário para resolver esses problemas. Porém, embora o modelo pareça ser capaz de emular etapas lógicas e matemáticas similares a uma pessoa, ainda há momentos em que comete erros graves em etapas triviais, atingindo menos de 40% de acerto no geral.

Nunes *et al.* (2023) realizaram um estudo similar aplicado ao contexto brasileiro: foi utilizado como base questões do Exame Nacional do Ensino Médio (ENEM), que serve como porta de entrada para várias universidades do Brasil. O ENEM apresenta um desafio único para os LLMs pois requer o entendimento profundo de vários campos do conhecimento e necessita

da habilidade de combinar informações de diversas áreas. Como diferencial dessa pesquisa temos o idioma, já que a maioria das avaliações dos Grandes Modelos de Linguagem é feita em tarefas escritas em inglês, enquanto nesse caso o conjunto dos dados está em português.

Para a pesquisa, os autores combinaram dois *datasets*. O primeiro, compilado por Silveira e Mauá (2008), contém originalmente 1745 questões do ENEM aplicadas entre 2009-2017. No entanto, perguntas que demandavam imagens para serem resolvidas, raciocínio matemático e entendimento de elementos químicos foram removidas, totalizando 916 ao fim. O segundo conjunto contém 180 questões do ENEM de 2022 reunidas pelos próprios autores. Seguindo o mesmo critério de remoção do conjunto anterior, ficaram 88. Utilizando somente o critério de exclusão por imagem, sobram 118.

Para avaliar as 118 questões do ENEM 2022 os autores usaram 3 modelos: code-davinci-002, gpt-3.5-turbo-0314 e gpt-4-0314, todos da OpenAI e disponíveis por meio de API paga. As técnicas de engenharia de prompt selecionadas foram *zero-shot*, *few-shot* e *few-shot with CoT* e os resultados estão separados por área na tabela a seguir.

Tabela 3 – Resultados do ENEM por área do conhecimento

Area	code-davinci-002			gpt-3.5-turbo			gpt-4		
	zero-shot	three-shot	three-shot with CoT	zero-shot	three-shot	three-shot with CoT	zero-shot	three-shot	three-shot with CoT
Languages and Codes	78.79	87.88	72.73	75.76	81.82	69.70	84.85	87.88	87.88
Human Sciences	89.19	94.59	91.89	91.89	89.19	94.59	94.59	94.59	94.59
Natural Sciences	69.23	61.54	53.85	73.08	84.62	65.38	84.62	76.92	88.46
Mathematics	18.18	27.27	50.00	18.18	36.36	54.55	40.91	50.00	72.73
Total	68.64	72.88	70.34	69.49	76.27	73.73	79.66	80.51	87.29

Fonte: Nunes *et al.* (2023)

Pela tabela podemos observar que as melhores métricas são atingidas pelo gpt-4 independentemente do método de *in-context learning* utilizado. Com zero-shot o modelo já atingiu uma acurácia de 79.66%, com desempenho excelente em Ciências Humanas (94.59%) e mediano em Matemática (40.91%). A partir do uso do CoT com 3 exemplos foi possível observar um salto de mais de 6 pontos percentuais (87.29%) considerando o resultado com todas as questões. As melhorias se deram especialmente em Matemática que foi para 72.73%. Olhando mais a fundo para os outros dois modelos, pode-se reparar que CoT também melhorou substancialmente os resultados de Matemática. Porém esses ganhos não são consistentes em Linguagem e Códigos, Ciências Humanas e Ciências Naturais e chegam a ter até o efeito

contrário, levando a uma queda na performance, o que evidencia que um *prompt* não é sempre melhor que outro em tarefas ou áreas do conhecimento distintas.

Como mencionado, o *dataset* contendo questões de 2009-2017 também foi analisado. A ideia era comparar as métricas novas com os melhores resultados atingidos até então e avaliar se os dados utilizados no aprendizado dos LLMs foram contaminados, ou seja, se um modelo como o gpt-3.5-turbo já viu questões do ENEM e suas respostas enquanto estava treinando e com isso pode ter memorizado a resposta ao invés de gerá-la de forma genuína.

Utilizando *zero-shot* foi possível alcançar uma média nesse período de 81.32% de acurácia com o gpt-3.5-turbo e 94.56% com o gpt-4. Se formos considerar somente as questões de 2022 temos 86.36% para o primeiro e 90.91% para o segundo. Embora não seja possível saber todos os detalhes desses modelos, temos a informação de que os dados para treinamento foram coletados até setembro de 2021. Levando essas informações em conta, os autores atingiram o novo estado da arte para esse problema, em comparação a 29.39% atingido por Silveira e Mauá (2008). Nunes *et al.* (2023) comentam que as métricas do gpt-4, que foram frequentemente melhores no período pré-2021, levantam certa preocupação pois podem indicar que esse modelo na verdade gravou parte das questões do ENEM durante a fase de treino.

4 METODOLOGIA

Para a realização dessa pesquisa a abordagem selecionada foi a quantitativa. Esta valoriza a mensuração e a análise estatística dos dados, permitindo a medição de opiniões, reações, hábitos e atitudes em um universo, por meio de uma amostra que o represente numericamente, de modo a generalizar os resultados (TERENCE; ESCRIVÃO FILHO, 2006).

Além da etapa da revisão bibliográfica para a elaboração da fundamentação teórica, a metodologia se constitui em seis outras etapas. São elas: coleta dos dados, seleção dos modelos a serem analisados, definição das técnicas de engenharia de *prompt* investigadas, criação de código para fazer a chamada dos modelos via API, tratamento dos dados retornados e a posterior análise dos resultados.

4.1 COLETA DE DADOS

Como fonte para a coleta de dados foram utilizados as provas e os gabaritos das últimas quatro edições de vestibular para ingresso no Instituto Militar de Engenharia (IME) e no Instituto Tecnológico da Aeronáutica (ITA).

Os dois institutos possuem duas fases de provas no processo de seleção. No IME, a primeira fase é composta somente de questões múltipla escolha de Matemática, Física e Química, enquanto a segunda fase possui Redação, questões objetivas de Português e Inglês e perguntas discursivas de Matemática, Física e Química. Já no ITA, a primeira fase é composta de questões de múltipla escolha de Matemática, Física, Química, Português e Inglês. A segunda fase é composta de Redação e questões discursivas de Matemática, Física e Química.

Foram coletadas somente as questões objetivas das disciplinas Matemática, Física, Química e Português, independentemente de estarem na primeira ou na segunda fase dos processos. As questões de Inglês não foram consideradas, pois o foco da pesquisa é justamente avaliar o desempenho dos modelos em resolver questões complexas que estejam escritas, assim

como as suas alternativas, em português. As provas e os gabaritos em formato pdf estão disponíveis online de forma aberta nos sites de vestibular das instituições^{1 2}.

No geral, o número mais frequente de questões coletadas por prova foi de 60, como pode-se ver na tabela 4. Uma pequena variação foi a prova de vestibular do ITA para acesso em 2023 em que houve a mudança de 15 para 12 questões por disciplina.

Tabela 4 – Distribuição das questões por prova e ano

Prova	Ano	Disciplina				Questões por prova
		Física	Matemática	Português	Química	
IME	2020	15	15	20	10	60
	2021	15	15	20	10	60
	2022	15	15	20	10	60
	2023	15	15	20	10	60
ITA	2020	15	15	15	15	60
	2021	15	15	15	15	60
	2022	15	15	15	15	60
	2023	12	12	12	12	48

Fonte: Elaborado pelo autor.

Vale notar que no IME as questões de Física e Matemática, 15 em ambas, e Química (10), que fazem parte da primeira fase da prova, possuem quantidades diferentes, enquanto as questões de Português que fazem parte da segunda etapa são 20. Já no ITA todas as disciplinas possuem a mesma quantidade de questões. Ao todo temos então 468 questões coletadas com 5 opções de resposta (A, B, C, D e E).

Dessas 468 questões, 108 foram removidas deste estudo. Destas 108, 77 (71.30%) foram desconsideradas por necessitarem da imagem presente no enunciado das questões ou nas alternativas para serem resolvidas, capacidade multimodal que os modelos selecionados nessa pesquisa não possuem. Outras 16 (14.81%) das 108 removidas necessitavam de mais espaço

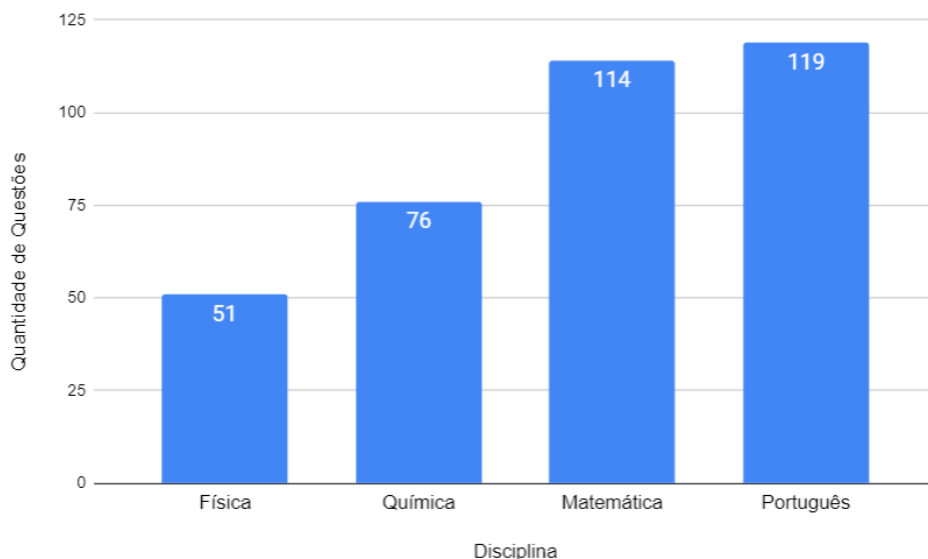
¹ INSTITUTO MILITAR DE ENGENHARIA (Brasil). **Provas anteriores do Concurso de Admissão para o CFG**. Rio de Janeiro: IME, 2020. Disponível em: <http://www.ime.eb.mil.br/provas-antiores-cfg.html>. Acesso em: 20 jul. 2023.

² INSTITUTO TECNOLÓGICO DE AERONÁUTICA (Brasil). **Provas anteriores**. São Paulo: ITA, 2023. Disponível em: <https://vestibular.ita.br/provas.htm>. Acesso em: 20 jul. 2023.

para texto em algumas configurações de *prompt* do que os parâmetros selecionados nos LLMs permitem e, por fim, as 15 (13.89%) restantes foram excluídas por terem sido anuladas de acordo com os gabaritos disponibilizados pelas bancas. Ao final, ficamos com uma amostra de 360 questões a serem respondidas.

Das 108, a maior parte das questões removidas foram de Física com 66 (61.11%), seguido por Química com 21 (19.44%), Português com 18 (16.67%) e Matemática com 3 (2.78%). Na figura 17, podemos observar o total de questões da amostra selecionada distribuídas por disciplina.

Figura 15 – Número de questões por disciplina



Fonte: Elaborado pelo autor.

Português é a área melhor representada com 33.06% das questões, seguida por Matemática com 31.67%, Química 21.11% e Física com 14.17%. O *dataset* com as questões está disponível no GitHub³ e possui as seguintes colunas:

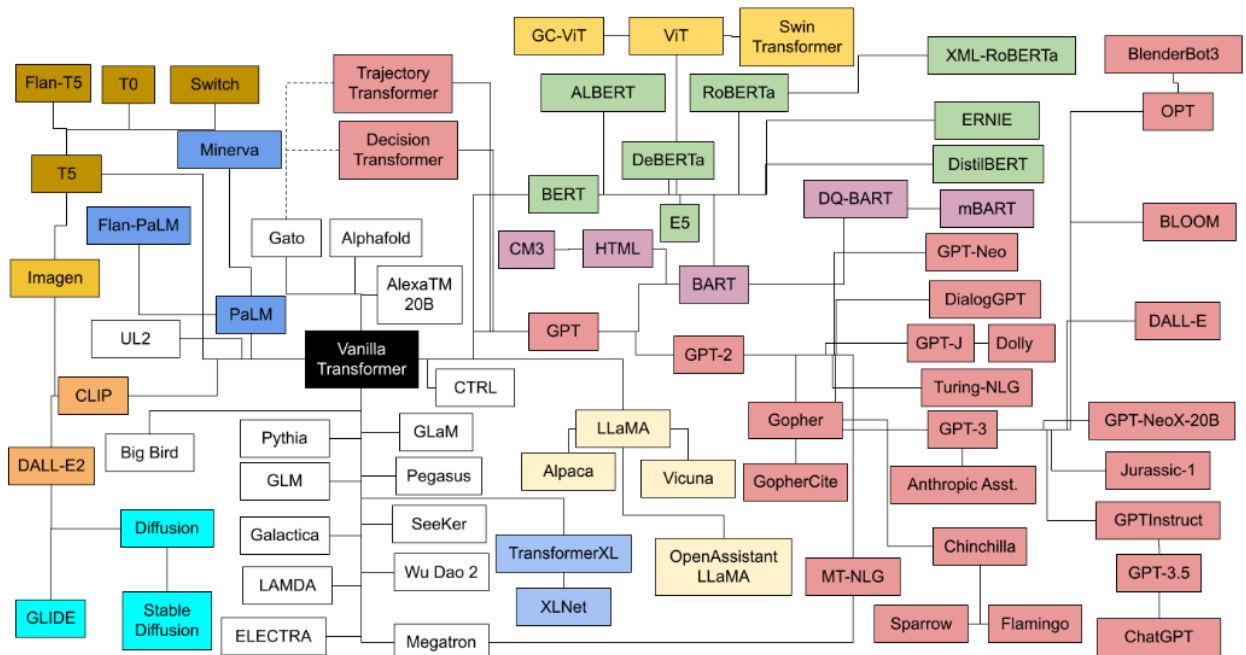
- id: Código de identificação único para cada questão;
- exam: Indicação de qual vestibular a questão pertence (IME ou ITA);
- year: Ano de entrada na instituição a partir do vestibular;

³ PERES, R. LLMs-IME_ITA: data. Rio de Janeiro, 2023. GitHub: @Rodrigo-Peres. Disponível em: https://github.com/Rodrigo-Peres/LLMs-IME_ITA/blob/main/data/IME_and_ITA_questions_dataset.xlsx. Acesso em: 02 ago. 2023.

- number: Número da questão na prova;
- question: Enunciado da questão;
- alternatives: Todas as alternativas múltipla escolha;
- test_answer: Letra da alternativa correta segundo o gabarito;
- subject: Disciplina da questão;
- status: Se a questão foi considerada para a pesquisa ou não;
- extra_info: Informação complementar para se responder uma questão, como um texto ou constante;
- few_shot: Com a solução de algumas questões utilizando a técnica de few-shot que vai servir de exemplo para outras perguntas da mesma disciplina;
- chain_of_thought: Com a solução de algumas questões utilizando a técnica de CoT que vai servir de exemplo para outras perguntas da mesma disciplina.

4.2 SELEÇÃO DOS MODELOS

Os primeiros modelos de linguagem baseados na tecnologia *Transformers*, como o GPT-1 (RADFORD *et al.*, 2018) e o BERT (DEVLIN *et al.*, 2019), foram desenvolvidos em 2018, mas somente com a implementação do ChatGPT (OPENAI, 2022) que aliou alta capacidade de resolução de problemas com uma interface agradável mesmo para pessoas pouco familiarizados com *Machine Learning* que essa tecnologia passou a receber enorme atenção. O ChatGPT atingiu números expressivos como 1 milhão de pessoas cadastradas em 5 dias (NOGUEIRA, 2023) e 100 milhões de usuários ativos em apenas 2 meses, se tornando a aplicação voltada para o público geral com o crescimento da base de usuários mais rápido da história até então (THE GUARDIAN, 2023).

Figura 16 – Modelos da Família *Transformers*

Fonte: Amatriain (2023)

Esses fatores fizeram com que outras organizações privadas e a comunidade *open source* se interessassem ainda mais por investirem e criarem ferramentas e modelos similares, como os da figura 16. Atualmente existem mais de 15.000 Grandes Modelos de Linguagem disponíveis para uso no HuggingFace, que é um repositório de modelos e conjuntos de dados (GAO; GAO, 2023)

Para essa pesquisa optou-se por utilizar somente LLMs disponíveis via API, dessa forma restringimos a busca aos modelos de 3 organizações: Anthropic, Google e OpenAI. No quadro 1 podemos ver informações sobre os modelos, as empresas responsáveis, número máximo de tokens aceitos, o tipo de acesso e se ele foi possível ou não, e a data de corte para a coleta dos dados usados para o treinamento.

Quadro 1 – Informações sobre os LLMs selecionados

Organização	Modelo	Máximo de tokens	Tipo de acesso	Acesso foi possível?	Corte dos dados de treinamento
OpenAI	text-davinci-003	4.097	API paga	Sim	06/2021
OpenAI	gpt-3.5-turbo-0301	4.096	API paga	Sim	09/2021
OpenAI	gpt-4-0314	8.192	API paga	Sim	09/2021
Google	text-bison@001	8.192	API paga e site	Sim	02/2023
Google	chat-bison@001	4.096	API paga e site	Sim	02/2023
Anthropic	claude-1.3	100.000	API paga	Não	Começo de 2023
Anthropic	claude-instant-1.1-100k	100.000	API paga	Não	Começo de 2023

Fonte: Elaborado pelo autor.

Os modelos da Anthropic, embora estejam disponíveis via API, não foram usados nessa pesquisa porque o acesso a eles está aberto somente para um número limitado de usuários no momento. O acesso foi solicitado, mas até o fim da etapa de coleta de dados nenhum retorno foi obtido.

Já os modelos text-bison@001 e chat-bison@001 também estão disponíveis via interface de usuário e por API através da plataforma de computação em nuvem Google Cloud Platform. Esses dois modelos também foram removidos da pesquisa por estarem em versão beta e no momento só estarem aceitando como entrada textos escritos em inglês.

Dessa forma selecionamos para o estudo três modelos da OpenAI, text-davinci-003, gpt-3.5-turbo e gpt-4, que cumpriram todos os requisitos de estarem disponíveis via API, sem barreiras para a criação de uma chave de acesso e que aceitam textos em português como entrada.

Os LLMs são treinados em enormes volumes de dados, como páginas de sites, e por isso existe o receio de que no momento da inferência esses modelos estejam na verdade lembrando as respostas corretas das perguntas que foram vistas na etapa de treinamento. Justamente para avaliar esse aspecto foram selecionadas, dos vestibulares de cada um dos institutos militares, duas provas aplicadas antes e duas aplicadas após o fim do período de coleta dos dados para treinamento dos modelos, que foi de junho a setembro de 2021.

Com relação aos parâmetros dos modelos, foram definidos os seguintes valores: *temperature* em 0, em linha com Nunes *et al.* (2023) e Arora, Singh e Mausam (2023), e *max_tokens* em 4096. Quanto menor o número da temperatura, mais o texto inferido será determinístico, por outro lado quanto maior, mais o texto predito será diverso e criativo (KIRK *et al.*, 2022). Já *max_tokens* controla o número de tokens que serão gerados na saída de um modelo, foi escolhido 4096 por ser o maior possível para igualar os LLMs selecionados.

4.3 ESCOLHA DAS TÉCNICAS DE *IN-CONTEXT LEARNING*

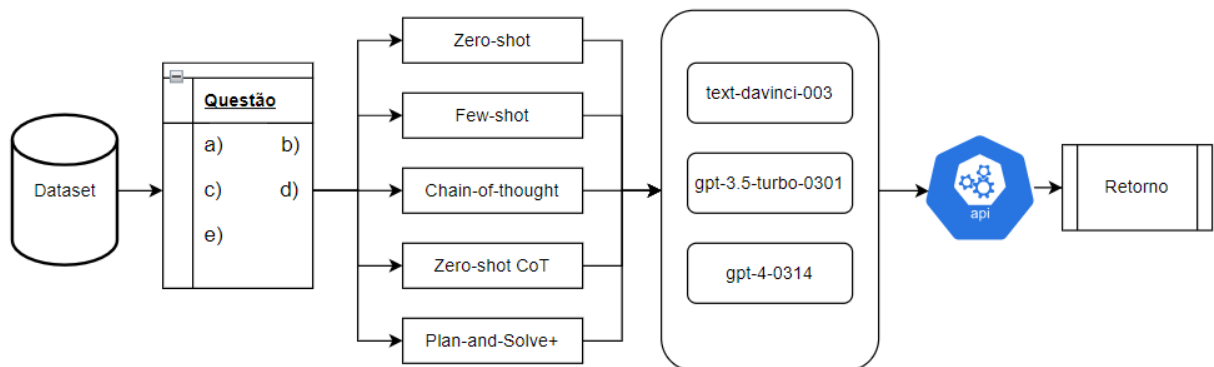
Existem atualmente várias técnicas de engenharia de *prompt* que visam extrair melhor desempenho dos Grandes Modelos de Linguagem. Para esse trabalho foram selecionados cinco métodos de *in-context learning*, todos eles consideradas de único turno e que foram detalhados no referencial teórico:

- *Zero-shot* (BROWN *et al.*, 2020);
- *Few-shot* (BROWN *et al.*, 2020);
- *Chain-of-Thought* (WEI *et al.*, 2022a);
- *Zero-shot Chain-of-Thought* (KOJIMA *et al.*, 2022);
- *Plan-and-Solve* (WANG *et al.*, 2023).

4.4 SOFTWARE PARA INTERAGIR COM OS MODELOS VIA API

Para obter as respostas dos modelos foi desenvolvido um software na linguagem Python e o seu funcionamento está demonstrado na imagem a seguir. Primeiro temos a leitura do conjunto de dados com todas as informações necessárias, é feito um *loop* onde cada questão é selecionada por vez junto das alternativas. Após isso temos a formatação do *prompt* de acordo com o método selecionado, depois temos a escolha do modelo e a definição dos parâmetros e o envio para a API, por fim recebemos o retorno e o salvamos em um novo arquivo.

Figura 17 – Ilustração do funcionamento do software



Fonte: Elaborado pelo autor.

O código e os *prompts* utilizados para executar os experimentos estão disponíveis no GitHub⁴, o que possibilita a reprodução. Para executá-lo é necessário ter uma chave de API da OpenAI e cada chamada possui um custo.

4.5 EXTRAÇÃO E TRATAMENTO DAS RESPOSTAS

Após os retornos dos modelos, na maioria dos casos respostas longas, esses textos são enviados novamente para o modelo gpt-3.5 em conjunto com alguns exemplos feitos utilizando a técnica *few-shot* e tem o objetivo de extrair somente a letra que representa a resposta correta, e assim facilitar a análise futura. Esses dados foram revisados manualmente para garantir a qualidade.

4.6 ANÁLISE DOS RESULTADOS

Ao todo 5 técnicas de engenharia de *prompt* foram testadas levando em conta os 3 modelos selecionados, totalizando 15 experimentos, os resultados das respostas foram analisados quantitativamente. Acurácia, que é a relação entre os acertos e o número total de

⁴ PERES, R. LLMs-IME_ITA. Rio de Janeiro, 2023. GitHub: @Rodrigo-Peres. Disponível em: https://github.com/Rodrigo-Peres/LLMs-IME_ITA/tree/main. Acesso em: 02 ago. 2023.

questões, foi a métrica selecionada. Embora os modelos tenham sido configurados com o menor valor de temperatura disponível visando atingir resultados determinísticos, também foram adicionados os desvios-padrão, calculados a partir de todos os acertos e erros que formam a acurácia, gerados com o uso de 100.000 amostras de *bootstrap* com reposição, para termos uma visão adicional dos resultados. Os dados foram investigados por diversos ângulos, sendo eles:

- Análise geral levando em conta todas as 360 questões;
- Análise dos resultados levando em conta as 4 provas ofertadas antes do fim da coleta dos dados usados no treinamento e as 4 provas aplicadas depois.
- Análise separada por disciplina, sendo elas Física, Matemática, Português e Química;
- Análise segregada por tipo de vestibular (IME e ITA);
- Análise da quantidade de acertos ou erros por questão;

5 RESULTADOS E DISCUSSÃO

Nesta seção serão apresentados e discutidos os resultados dos experimentos executados.

5.1 RESULTADOS GERAIS

Na tabela 5 presente na página a seguir temos os resultados gerais alcançados pela combinação dos três modelos em conjunto com as cinco técnicas de *in-context learning* selecionadas para a pesquisa, totalizando quinze experimentos. Além dessas separações, temos a acurácia geral, dando o mesmo peso para todas as questões e a acurácia ponderada, onde é aplicado peso diferenciado nas questões visando equilibrar o número de perguntas por disciplina, isso se faz necessário pois temos por exemplo 51 questões de Física e 119 de Português. Adicionalmente incluímos um baseline que seria o resultado alcançado de forma aleatória simples e o desvio-padrão calculado com *bootstrap*.

Tabela 5 – Resultados gerais por modelo e técnica de engenharia de *prompt*

		text-davinci-003										
		Aleatório	zero-shot	DP	zero-shot CoT	DP	PS+	DP	few-shot	DP	CoT	DP
Acurácia total		20.00	35.00	2.52	33.61	2.52	36.94	2.52	38.05	2.54	35.00	2.49
Acurácia ponderada		20.00	34.25	2.66	33.62	2.63	37.17	2.69	38.04	2.71	34.15	2.66
		gpt-3.5-turbo-0301										
		Aleatório	zero-shot	DP	zero-shot CoT	DP	PS+	DP	few-shot	DP	CoT	DP
Acurácia total		20.00	35.00	2.50	34.16	2.48	22.77	2.23	38.61	2.55	39.16	2.56
Acurácia ponderada		20.00	34.52	2.64	33.86	2.64	23.53	2.35	38.34	2.71	38.47	2.72
		gpt-4-0314										
		Aleatório	zero-shot	DP	zero-shot CoT	DP	PS+	DP	few-shot	DP	CoT	DP
Acurácia total		20.00	48.05	2.64	47.22	2.62	46.11	2.61	48.61	2.63	55.00	2.63
Acurácia ponderada		20.00	48.52	2.78	48.14	2.78	45.39	2.77	46.60	2.80	55.04	2.76

Fonte: Elaborado pelo autor.

Com relação aos dados do primeiro modelo elencado, *text-davinci-003*, temos como destaque que a técnica de *zero-shot*, considerada a mais simples, não ficou distante em relação às outras. Vale mencionar que esse modelo é mais antigo e foi o primeiro dos três a ser lançado. Esses resultados indicam que técnicas que visam uma elaboração maior da resposta e que tenham o objetivo de gerar etapas de raciocínio não tiveram muito efeito. O melhor resultado de acurácia total (38.05) e ponderada (38.03) foi da técnica de *few-shot*, onde são passados exemplos em conjunto com as respostas, sem a cadeia de pensamento. *Zero-shot* atingiu 35.00 e 34.25, diferença de ao menos 3.05 pontos percentuais. Considerando os valores de desvio-padrão, estas diferenças não se apresentam tão significantes.

O caso do *gpt-3.5-turbo-0301* é diferente, esse modelo é uma evolução do *text-davinci-003* e é otimizado para conversação (OPENAI, c2023a). O resultado *zero-shot* foi em linha com o modelo anterior. Como destaque positivo tivemos novamente a técnica *few-shot* com 38.61 de acurácia geral e *chain-of-thought* sendo o melhor com 39.16 no geral e 38.47 de acurácia ponderada, uma diferença de ao menos 3.95 em relação ao *zero-shot*. Essa técnica tem a proposta de não só enviar exemplos com a resposta, mas também o passo a passo para chegar nela. Como destaque negativo tivemos o uso da técnica *plan and solve+*, que obteve um resultado próximo do aleatório.

Por fim, no *gpt-4-0314*, uma evolução do *gpt-3.5*, foi possível observar que mesmo com técnicas mais simples atingiu-se uma taxa de acerto de 45% em todos os cenários, superando expressivamente os anteriores. Já com relação às estratégias de engenharia de *prompt*, os resultados ficaram em linha com o modelo anterior. O *chain-of-thought* também se destacou em um modelo que passou pela etapa de *fine-tuning* para ter boa performance em conversação (OPENAI, c2023b). Com essa técnica foi possível atingir o melhor resultado geral com 55.00, assim como levando em conta a métrica da acurácia ponderada com 55.04, ao menos 6.52 pontos percentuais de diferença em relação a *zero-shot* e o único experimento capaz de acertar mais questões do que errar.

No geral, podemos concluir que existe um salto de qualidade entre o *gpt-4* e os outros modelos mais antigos. Os melhores resultados foram atingidos por modelos otimizados para conversação em conjunto com *chain-of-thought*. Em todos os cenários, mesmo utilizando *zero-shot*, os resultados foram melhores do que a simulação da marcação aleatória das respostas. Outro ponto a se destacar é que separação entre as métricas de acurácia geral e ponderada fez pouca diferença em relação às colocações dos experimentos, o único caso foi o do *gpt-4* com

few-shot em que obteve 48.61 contra 46.60, o que indica que o resultado geral foi influenciado demasiadamente pela performance em uma disciplina.

Com relação a visão com o desvio-padrão, temos dois grandes destaques, o negativo vai para a combinação de gpt-3.5-turbo-0301 com *plan and solve+*, disparado as piores acurácias mesmo levando em conta os intervalos. Já o destaque positivo fica com a combinação do gpt-4-0314 com *chain-of-thought*, que lidera com folga em relação aos outros experimentos.

5.2 RESULTADOS POR DISCIPLINA

A seguir temos a tabela 6 com os mesmos resultados da seção anterior só que destrinchados entre as disciplinas de Física, Química, Matemática e Português.

Tabela 6 – Resultados por disciplina

text-davinci-003											
	Aleatório	zero-shot	DP	zero-shot CoT	DP	PS+	DP	few-shot	DP	CoT	DP
Física	20.00	33.33	6.59	35.29	6.69	45.09	6.95	37.25	6.77	31.37	6.49
Química	20.00	28.94	5.21	32.89	5.38	28.94	5.21	40.78	5.66	30.26	5.26
Matemática	20.00	21.05	3.80	18.42	3.64	22.80	3.92	23.68	3.99	23.68	3.98
Português	20.00	52.94	4.58	47.89	4.59	52.10	4.58	50.42	4.56	50.42	4.60
gpt-3.5-turbo-0301											
	Aleatório	zero-shot	DP	zero-shot CoT	DP	PS+	DP	few-shot	DP	CoT	DP
Física	20.00	31.37	6.47	33.33	6.59	23.52	5.93	39.21	6.83	37.25	6.76
Química	20.00	36.84	5.53	32.89	5.42	32.89	5.39	35.52	5.48	34.21	5.44
Matemática	20.00	11.40	2.98	21.05	3.81	15.78	3.40	24.56	4.03	25.43	4.08
Português	20.00	57.98	4.52	47.89	4.57	22.68	3.82	53.78	4.58	56.30	4.54
gpt-4-0314											
	Aleatório	zero-shot	DP	zero-shot CoT	DP	PS+	DP	few-shot	DP	CoT	DP
Física	20.00	47.05	7.00	52.94	7.01	41.17	6.89	33.33	6.60	50.98	7.00
Química	20.00	59.21	5.63	53.94	5.71	46.05	5.73	47.36	5.71	64.47	5.49
Matemática	20.00	21.92	3.86	21.05	3.82	28.94	4.23	28.07	4.20	33.33	4.42
Português	20.00	66.38	4.33	65.54	4.33	64.70	4.38	75.63	3.91	71.42	4.14

Fonte: Elaborado pelo autor.

Com o modelo *text-davinci-003*, observamos uma performance semelhante em Português e Matemática entre as diversas técnicas de engenharia de *prompt*, embora em Matemática todas as predições tenham sido estatisticamente iguais ao resultado aleatório. Já nas outras disciplinas as técnicas de *prompt* que se saíram melhor foram aquelas que enviam exemplos ou geram o passo a passo para resolução das questões: Física com PS+ (45.09) e Química (40.78) com *few-shot*.

Para o *gpt-3.5-turbo-0301*, o resultado de Português está em linha com o modelo anterior, exceto para a técnica de PS+. Em Matemática, as melhores notas foram atingidas por *zero-shot CoT*, *few-shot* e CoT. Para as outras disciplinas, tivemos Física com *few-shot* (39.21) e CoT e Química com *zero-shot* (36.84), resultado próximo de *few-shot* (35.52) e CoT (34.21). Contudo, cabe ressaltar que considerando os desvios-padrões estas diferenças tendem a ser menos significantes estatisticamente.

Para o *gpt-4-0314*, temos destaque para a técnica CoT, sendo possível atingir as melhores notas. Mesmo levando em conta todas as combinações de modelos e técnicas de engenharia de *prompt* empregadas anteriormente o modelo obteve em Matemática (33.33) e Química (64.47) e também as segundas melhores pontuações em Física (50.98), atrás de *zero-shot CoT* (52.94), e Português (71.42), atrás de *few-shot* (75.63).

Por conta da quantidade de questões ser menor devido à quebra por disciplina, os valores de desvio-padrão ficaram elevados. Como destaque, temos o modelo *gpt-3.5-turbo-0301* em conjunto com PS+ aplicado em questões de Português que foi disparado o pior resultado geral nessa disciplina, afetando também a acurácia geral dessa união. Como demonstrado na seção 2.5.4 a instrução da técnica PS+ é enviesada para problemas matemáticos, a partir disso esse resultado ruim pode ser explicado pela recusa dessa combinação de modelo e *prompt* em resolver boa parte dos problemas de Português, ao todo 64 dos 119 (53.78 %), com a frase “Desculpe, mas não consigo resolver essa questão específica” aparecendo 17 vezes entre as negativas. Ainda sobre PS+ nessa disciplina, esse comportamento apareceu em menor grau no *gpt-4-0310* em 17 questões (14.28 %). No *text-davinci-003* esse ponto foi ignorado e todas as 119 perguntas tiveram uma resposta válida.

5.3 RESULTADOS POR ANO

Nesta seção usaremos a tabela a seguir para observar se existe o problema da contaminação dos dados usados para treinamento, ou seja, se algum dos modelos pode ter “visto” parte das questões e suas respostas durante o seu processo de aprendizagem.

Tabela 7 – Resultados por ano

		text-davinci-003									
		zero-shot	DP	zero-shot CoT	DP	PS+	DP	few-shot	DP	CoT	DP
	2020	35.16	5.00	37.36	5.06	38.46	5.08	35.16	4.98	30.77	4.83
	2021	35.64	4.75	37.62	4.81	37.62	4.82	37.62	4.83	33.66	4.72
	Acurácia pré-treinamento	35.42	3.44	37.50	3.49	38.02	3.50	36.46	3.45	32.29	3.37
	2022	29.17	4.62	27.08	4.55	34.38	4.86	40.63	5.00	32.29	4.76
	2023	41.67	5.81	31.94	5.47	37.50	5.71	38.89	5.75	45.83	5.85
	Acurácia pós-treinamento	34.52	3.67	29.17	3.49	35.71	3.69	39.88	3.78	38.10	3.74
		gpt-3.5-turbo-0301									
		zero-shot	DP	zero-shot CoT	DP	PS+	DP	few-shot	DP	CoT	DP
	2020	37.36	5.05	34.07	4.97	23.08	4.41	37.36	5.07	38.46	5.09
	2021	35.64	4.76	28.71	4.49	28.71	4.48	34.65	4.73	32.67	4.65
	Acurácia pré-treinamento	36.46	3.62	31.25	3.35	26.04	3.17	35.94	3.47	35.42	3.43
	2022	31.25	4.73	35.42	4.86	20.83	4.14	36.46	4.89	43.75	5.05
	2023	36.11	5.65	40.28	5.78	16.67	4.40	48.61	5.88	43.06	5.83
	Acurácia pós-treinamento	33.33	3.47	37.50	3.73	19.05	3.03	41.67	3.80	43.45	3.82
		gpt-4-0314									
		zero-shot	DP	zero-shot CoT	DP	PS+	DP	few-shot	DP	CoT	DP
	2020	51.65	5.24	51.65	5.24	53.85	5.22	53.85	5.22	56.04	5.20
	2021	43.56	4.94	47.52	4.97	42.57	4.92	38.61	4.83	51.49	4.97
	Acurácia pré-treinamento	47.40	3.59	49.48	3.60	47.92	3.59	45.83	3.61	53.65	3.60
	2022	51.04	5.07	45.83	5.07	41.67	5.01	48.96	5.09	52.08	5.10
	2023	45.83	5.85	43.06	5.82	47.22	5.86	55.56	5.85	62.50	5.71
	Acurácia pós-treinamento	48.81	3.84	44.64	3.84	44.05	3.80	51.79	3.87	56.55	3.82

Fonte: Elaborado pelo autor.

De acordo com os dados de acurácia obtidos e observando os desvios não é possível observar uma tendência clara que leve a apontar a memorização das questões durante o treinamento, em 8 dos 15 experimentos o resultado dos anos mais recentes foram melhores em relação ao período anterior. Isso pode indicar que esses dados não estão presentes em larga escala na internet, como as perguntas do ENEM, e que mesmo que as questões estejam disponíveis, as respostas não, o que diminui a chance desse fenômeno de “memorização” acontecer.

5.4 RESULTADOS POR INSTITUTO

Na sequência, temos a análise dos experimentos por modelo e técnica de engenharia de *prompt* em relação às provas de vestibular de cada um dos institutos militares.

Tabela 8 – Resultados por prova

		IME	DP	ITA	DP
text-davinci-003	zero-shot	31.55	3.58	38.02	3.49
	zero-shot CoT	32.14	3.61	34.90	3.46
	PS+	36.31	3.71	37.50	3.48
	few-shot	39.88	3.78	36.46	3.47
	CoT	38.10	3.73	32.29	3.38
gpt-3.5-turbo-0301	zero-shot	30.36	3.55	39.06	3.52
	zero-shot CoT	30.95	3.57	36.98	3.48
	PS+	20.83	3.14	24.48	3.11
	few-shot	35.71	3.70	41.15	3.54
	CoT	39.88	3.76	38.54	3.51
gpt-4-0314	zero-shot	46.43	3.85	49.48	3.60
	zero-shot CoT	43.45	3.82	50.52	3.60
	PS+	45.24	3.83	46.88	3.60
	few-shot	45.83	3.85	51.04	3.60
	CoT	50.60	3.85	58.85	3.55

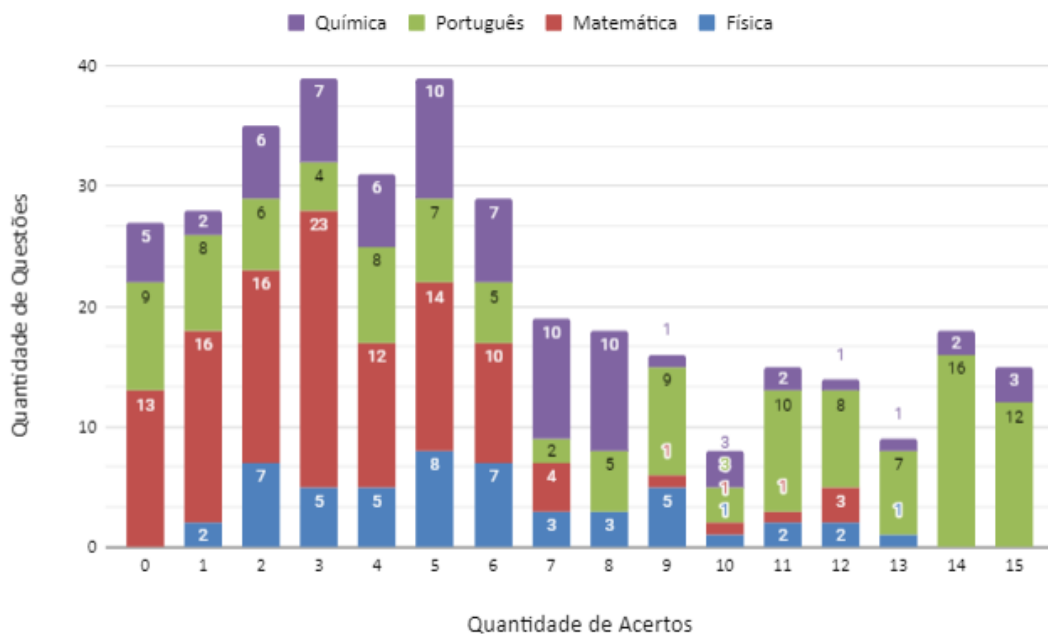
Fonte: Elaborado pelo autor.

Dos 15 experimentos observados, em 12 foi possível obter resultados melhores nas questões do ITA e em somente 3 as métricas foram melhores na prova do IME. Isso sugere que para os modelos essas questões apresentaram um nível de dificuldade ou complexidade menor, o que também pode representar um tipo diferente de abstração para se resolver as perguntas.

5.5 RESULTADOS EM RELAÇÃO A ACERTOS E ERROS

Na figura 18, temos a quantidade de acertos por questão considerando os 15 experimentos. O número de questões está subdividido entre as áreas do conhecimento analisadas: Física, Química, Matemática e Português.

Figura 18 – Quantidade de questões certas por Disciplina



Fonte: Elaborado pelo autor.

De acordo com a figura 18, podemos observar que temos uma concentração maior de questões que os modelos resolveram incorretamente, com até 6 respostas corretas entre 15 possíveis. Já a partir dos 7 acertos esse número se reduz e em nenhuma situação daí em diante passamos da barreira das 20 questões. Com relação às disciplinas podemos observar que o número de acertos cai drasticamente para as questões mais relacionadas a área de exatas como Física, Matemática e Química, sendo somente a última com alta quantidade de acertos, 14 e 15

em algumas questões (5). Português possui o comportamento contrário com uma quantidade relevante de questões (28) concentradas na área a partir de 14 acertos, o que demonstra que estas são mais apropriadas às técnicas de *prompt* ou ao menos que requerem um entendimento já consolidado nos modelos utilizados.

A seguir temos alguns exemplos de perguntas e respostas retornadas pelos experimentos. No quadro 2 temos uma questão de Português e no quadro 3 temos uma questão de Química, esses dois casos são de questões que obtiveram todas as respostas corretas, independente do modelo ou da técnica de engenharia de *prompt* utilizada. Vale destacar que para Química, as 5 questões com alto grau de acerto são relacionadas a conteúdo teórico e nenhum cálculo avançado é necessário para resolvê-las, como exemplificado no quadro 3.

Já nos quadros 4 e 5 temos exemplos de questões onde todas as respostas foram incorretas. No primeiro temos um caso de Português com o seguinte enunciado: Assinale a alternativa em que as palavras recebem o acento gráfico de acordo com as mesmas regras de acentuação das palavras abaixo transcritas, respectivamente: LÍDER (texto 1, ref. 6), INCLUÍDO (texto 1, ref. 7). Das 15 respostas incorretas 12 foram exatamente na mesma opção, alternativa “c) líderes, difícil”, o que parece indicar que a palavra “líder” no enunciado da questão pode ter enviesado as respostas geradas. O quadro 5 traz um caso de Matemática que necessita de cálculo para ser concluído, nele mesmo os modelos mais avançados, como o gpt-4-0314, com auxílio de *chain-of-thought*, apresentam dificuldade de executar os passos intermediários para chegar no resultado correto, aqui a alternativa a) 144.

Quadro 2 – Questão de Português com todas as respostas corretas

modelo	prompt	questão	alternativas	resposta
text-davinci-003	few_shot	"Ouve-se uma voz de comando; e um turbilhão de balas rola estrugidoramente dentro das galhadas..." (Texto 1, linhas 37 e 38). O valor semântico do vocábulo "estrugidoramente" no trecho acima se aproxima de:	a) violentamente. b) ruidosamente. c) velozmente. d) certamente. e) mortalmente.	A resposta correta é a letra b) ruidosamente.
text-davinci-003	zero_shot			b) ruidosamente.
gpt-4-0314	chain_of_thought			O termo "estrugidoramente" tem origem no verbo "estrugir", que significa produzir um ruído forte, estrondo. Portanto, a alternativa que mais se aproxima do valor semântico de "estrugidoramente" é a letra b) ruidosamente.
gpt-3.5-turbo-0301	chain_of_thought			O vocábulo "estrugidoramente" indica um som alto e intenso, portanto, seu valor semântico se aproxima de "ruidosamente". A resposta correta é a letra b) ruidosamente.

Fonte: Elaborado pelo autor.

Quadro 3 – Questão de Química com todas as respostas corretas

modelo	prompt	questão	alternativas	resposta
text-davinci-003	few_shot	Um professor de química propôs, como primeira etapa do mecanismo de esterificação do terc-butanol com o ácido acético, a formação de um carbocátion terciário no álcool. Suponha a viabilidade dessa proposta. O átomo do ácido acético mais propenso a realizar o ataque nucleofílico ao carbocátion formado seria o:	a) oxigênio do grupo hidroxila, pois seria o átomo mais eletronegativo por estar ligado a um átomo de hidrogênio. b) oxigênio da carbonila, pois facilmente assume uma carga negativa formal por ressonância. c) carbono do grupo ácido, pois facilmente assume a forma de carbânion por deslocamento de carga eletrônica. d) carbono do grupo metila, pois é o menos impedido espacialmente entre os dois carbonos. e) hidrogênio do grupo hidroxila, pois consegue se dissociar e formar um hidreto, um dos compostos mais eletronegativos existentes.	A resposta correta é a letra b) oxigênio da carbonila, pois facilmente assume uma carga negativa formal por ressonância.
gpt-4-0314	zero_shot			b) oxigênio da carbonila, pois facilmente assume uma carga negativa formal por ressonância.
gpt-4-0314	chain_of_thought			A resposta correta é a letra b) oxigênio da carbonila, pois facilmente assume uma carga negativa formal por ressonância. O oxigênio da carbonila é o átomo mais propenso a realizar o ataque nucleofílico ao carbocátion formado, pois é capaz de estabilizar a carga negativa através da ressonância. Isso ocorre porque o oxigênio é um átomo altamente eletronegativo e, portanto, é capaz de suportar uma carga negativa. Além disso, a estrutura de ressonância do ácido acético permite que a carga negativa seja distribuída entre os dois átomos de oxigênio, o que aumenta a estabilidade do íon.
gpt-3.5-turbo-0301	chain_of_thought			A proposta do professor é viável, pois a formação de um carbocátion terciário é possível na esterificação do terc-butanol com ácido acético. O átomo do ácido acético mais propenso a realizar o ataque nucleofílico ao carbocátion formado seria o oxigênio da carbonila, pois ele pode facilmente assumir uma carga negativa formal por ressonância, o que aumenta sua capacidade nucleofílica. Portanto, a alternativa correta é a letra b) oxigênio da carbonila.

Fonte: Elaborado pelo autor.

Quadro 4 – Questão de Português com todas as respostas incorretas

modelo	prompt	questão	alternativas	resposta
gpt-4-0314	chain_of_thought	Assinale a alternativa em que as palavras recebem o acento gráfico de acordo com as mesmas regras de acentuação das palavras abaixo transcritas, respectivamente: LÍDER (texto 1, ref. 6), INCLUÍDO (texto 1, ref. 7)	a) ânsia, balaústre. b) bíceps, ciúme. c) líderes, difícil. d) país, ciúme. e) fácil, fiéis.	A resposta correta é a letra c) líderes, difícil.
text-davinci-003	zero_shot			C) líderes, difícil.
gpt-3.5-turbo-0301	chain_of_thought			As palavras "líder" e "incluído" são acentuadas de acordo com a regra de acentuação das palavras oxítonas terminadas em "r" e "i", respectivamente. A única alternativa que apresenta palavras com a mesma regra de acentuação é a letra c) líderes, difícil.

Fonte: Elaborado pelo autor.

Quadro 5 – Questão de Matemática com todas as respostas incorretas

modelo	prompt	questão	alternativas	resposta
gpt-4-0314	chain_of_thought	<p>Seja a equação do terceiro grau em x:</p> $x^3 + p_1x^2 + p_2x + p_3 = 0$ <p>Onde $p_1 < p_2 < p_3$ são números primos menores que 100. Para que a razão entre a soma e o produto das raízes da equação seja a maior possível, o valor de $p_2 + p_3$ deve ser:</p>	<p>a) 144. b) 152. c) 162. d) 172. e) 196.</p>	<p>Pela relação de Girard, temos que a soma das raízes é $-p_1$, o produto das raízes é p_3 e a soma dos produtos das raízes tomadas duas a duas é p_2. Queremos maximizar a razão entre a soma e o produto das raízes, ou seja, maximizar $-p_1/p_3$.</p> <p>Como $p_1 < p_2 < p_3$, e todos são primos menores que 100, os maiores valores possíveis para p_1, p_2 e p_3 são 89, 97 e 101, respectivamente. No entanto, 101 não é menor que 100, então o maior valor possível para p_3 é 97. Portanto, a maior razão $-p_1/p_3$ ocorre quando $p_1 = 89$ e $p_3 = 97$, e a razão é $-89/97$.</p> <p>Agora, queremos encontrar $p_2 + p_3$. Como $p_2 < p_3$ e $p_3 = 97$, o maior valor possível para p_2 é 89. Portanto, $p_2 + p_3 = 89 + 97 = 186$.</p> <p>No entanto, 186 não está entre as alternativas. Vamos verificar se há algum erro em nossa análise. Note que p_1, p_2 e p_3 são todos primos e $p_1 < p_2 < p_3$. Além disso, $p_1 + p_2 + p_3$ é a soma das raízes, que deve ser negativa (pois a soma das raízes é $-p_1$). Portanto, não podemos ter $p_1 = 89$, $p_2 = 89$ e $p_3 = 97$, pois a soma das raízes seria positiva.</p> <p>Vamos tentar outros valores de p_1, p_2 e p_3. Se $p_1 = 83$, $p_2 = 89$ e $p_3 = 97$, a soma das raízes seria -83, que é negativa. Nesse caso, $p_2 + p_3 = 89 + 97 = 186$, que ainda não está entre as alternativas.</p> <p>Se $p_1 = 79$, $p_2 = 83$ e $p_3 = 89$, a soma das raízes seria -79, que é negativa. Nesse caso, $p_2 + p_3 = 83 + 89 = 172$, que está entre as alternativas.</p> <p>Portanto, o valor de $p_2 + p_3$ deve ser 172. A resposta correta é a letra d) 172.</p>

Fonte: Elaborado pelo autor.

6 CONSIDERAÇÕES FINAIS

Este trabalho investigou o desempenho dos Grandes Modelos de Linguagem na resolução de problemas de vestibulares de instituições militares brasileiras, especificamente o Instituto Militar de Engenharia (IME) e o Instituto Tecnológico da Aeronáutica (ITA). Através da coleta e preparação de um novo conjunto de dados contendo questões desses exames escritas em português, foi possível realizar experimentos controlados para avaliar diferentes abordagens.

Os resultados indicam que, de maneira geral, os LLMs mais recentes e otimizados para conversação obtiveram melhores resultados na resolução das questões, especialmente quando empregadas técnicas de *in-context learning* como o *chain-of-thought*. O modelo gpt-4 destacou-se alcançando 55% de acurácia geral com essa técnica de prompt, superando os modelos mais antigos e sendo o único experimento com mais acertos do que erros.

Ao analisar os resultados por disciplina nos modelos mais antigos, verificou-se que para Português os melhores desempenhos foram alcançados com técnicas simples, como o *zero-shot*, enquanto para Matemática, Física e Química se destacaram os métodos que buscam eliciar o raciocínio. Já no gpt-4, modelo mais avançado, tivemos destaque novamente para *chain-of-thought*, onde fazemos modificações no *prompt* para incluir não só exemplos, mas também cadeias de pensamento que levam à conclusão da questão. Não foi identificada nenhuma indicação clara de memorização das perguntas e de suas respostas pelos modelos durante o treinamento.

As questões do vestibular do ITA se mostraram um pouco menos desafiadoras para os LLMs do que as do IME, resultado que pode indicar diferenças no nível de complexidade e abstração requerido para a resolução.

De maneira geral, o trabalho explorou o potencial dos Grandes Modelos de Linguagem na resolução de problemas, com destaque positivo para as questões de Português e negativo para Matemática. Os resultados alcançados colocam em questão os benefícios e limitações atuais desses modelos, fornecendo evidências valiosas para pesquisas futuras voltadas ao aprimoramento e adequada implementação dos *Large Language Models* em contextos educacionais.

Apesar dos resultados alcançados, este trabalho apresenta algumas limitações que poderiam ser exploradas em pesquisas futuras. Primeiramente, foram utilizados apenas três modelos pré-treinados, todos da OpenAI, sendo que a adição de uma gama mais ampla de LLMs, incluindo aqueles de acesso fechado de outras companhias ou vindos da comunidade *open source*, poderia agregar à pesquisa.

Além disso, neste estudo foram empregadas somente técnicas de único turno de engenharia de *prompt*, como *zero-shot* e *chain-of-thought*. A investigação de abordagens mais avançadas de *prompting*, como *Self-Refine* (MADAAN *et al.*, 2022) e *Self-Consistency* (WANG *et al.*, 2022), consideradas de múltiplos turnos (KADDOUR *et al.*, 2023), pode contribuir para melhorar os desempenhos alcançados. Outro aspecto relevante é que os LLMs são majoritariamente treinados em inglês, sendo escassos os dados de treinamento em português. Modelos treinados especificamente na língua portuguesa podem apresentar melhor adaptação ao idioma e conseqüentemente às questões (PIRES *et al.*, 2022).

Ainda, apesar do *chain-of-thought* ter se destacado, este não conseguiu atingir resultados ótimos. As etapas de raciocínio geradas pelo modelo podem conter inconsistências, principalmente quando relacionadas a cálculos avançados. Portanto, é necessário aprofundar uma avaliação qualitativa para verificar a confiabilidade dos passos produzidos antes de utilizá-los para fins educacionais. Por fim, a formatação das questões, tanto em português quanto em notação matemática LaTeX, pode impactar o desempenho e é um outro caminho a ser investigado (ETXANIZ *et al.*, 2023).

A pesquisa em aplicações educacionais em Grandes Modelos de Linguagem está embrionária, tendo enorme potencial em diversas direções, sendo esse trabalho uma modesta contribuição a esse vasto campo que está em rápida evolução. À medida que as investigações avançam, novas abordagens, modelos e técnicas certamente surgirão no sentido de beneficiar professores e a aprendizagem de estudantes em todo o mundo.

REFERÊNCIAS

- ALMEIDA, T. S. *et al.* BLUEX: A benchmark based on Brazilian Leading Universities Entrance eXams. **arXiv** [preprint], p. 1-9, 11 July 2023. arXiv:2307.05410v1.
- ALOM, M. Z. *et al.* The history began from AlexNet: a comprehensive survey on deep learning approaches. **arXiv** [preprint], p. 1-39, 12 Sept. 2018. arXiv:1803.01164.
- ALZUBAIDI, L. *et al.* Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. **Journal of Big Data**, v. 8, n. 1, 2021.
- AMATRIAIN, X. Transformer models: an introduction and catalog. **arXiv** [preprint], p. 1-67, 25 May 2023. arXiv:2302.07730v3.
- ARAÚJO, L. R. de; MATOS, T. G. R. Expectativas e vivências acadêmicas de alunos de instituições militares do ensino superior. *In: ENCONTRO DE PÓS-GRADUAÇÃO E PESQUISA*, 21., 2021, Fortaleza. **Anais** [...]. Fortaleza: Unifor, 2021.
- ARORA, D.; SINGH, H.; MAUSAM. Have LLMs Advanced Enough? A Challenging Problem Solving Benchmark For Large Language Models. **arXiv** [preprint], p. 1-9, 24 May 2023. arXiv:2305.15074.
- BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. **arXiv** [preprint], p. 1-15, 1 Sept. 2014. arXiv:1409.0473.
- BOWMAN, S. R. Eight things to know about large language models. **arXiv** [preprint], p. 1-16, 2 Apr. 2023. arXiv:2304.00612.
- BROWN, T. *et al.* Language models are few-shot learners. *In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, 33., 2020, [virtual]. **Proceedings** [...]. Vancouver, 2020. p. 1-25.
- CHOWDHARY, K. R. **Fundamentals of artificial intelligence**. New Delhi: Springer India, 2020.
- CHUNG, J. *et al.* (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. **arXiv** [preprint], p. 1-9, 11 Dec. 2014. arXiv:1412.3555v1.
- CORNEGRUTA, S. *et al.* Modelling radiological language with bidirectional long short-term memory networks. *In: SEVENTH INTERNATIONAL WORKSHOP ON HEALTH TEXT MINING AND INFORMATION ANALYSIS*, 2016, Austin. **Proceedings** [...]. Austin: ACL, 2016. p. 17-27.
- DEVLIN, J. *et al.* BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *In: CONFERENCE OF THE NORTH AMERICAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS: HUMAN LANGUAGE TECHNOLOGIES*, 2019, Minneapolis. **Proceedings** [...]. Minneapolis: ACL, 2019. p. 4171-4186.

ETXANIZ, J. *et al.* Do Multilingual Language Models Think Better in English? **arXiv** [preprint], p. 1-15, 02 Aug. 2023. arXiv:2308.01223.

GAO, S.; GAO, A. K. On the origin of LLMs: an evolutionary tree and graph for 15,821 Large Language Models. **arXiv** [preprint], p. 1-14, 19 July 2023. arXiv:2307.09793.

GUNASEKAR, S. *et al.* Textbooks Are All You Need. **arXiv** [preprint], p. 1-26, 20 Jun 2023. arXiv:2306.11644.

HOCHREITER, S.; SCHMIDHUBER, J. Long Short-Term Memory. **Neural Computation**, Cambridge, v. 9, n. 8, p. 1735-1780, 1997.

KADDOUR, J. *et al.* Challenges and applications of Large Language Models. **arXiv** [preprint], p. 1-72, 19 July 2023. arXiv:2307.10169.

KAPLAN, J. *et al.* Scaling laws for neural language models. **arXiv** [preprint], p. 1-30, 23 Jan. 2020. arXiv:2001.08361.

KASNECI, E. *et al.* ChatGPT for good? On opportunities and challenges of large language models for education. **Learning and Individual Differences**, v. 103, p. 1-13, 2023.

KIRK, J. *et al.* Improving Language Model Prompting in Support of Semi-autonomous Task Learning. **Advances in Cognitive Systems**, v. 10, p. 1-18, 2022.

KLEIN, D. **CS 294-5**: statistical natural language processing: course introduction, lecture I. Berkeley: Electrical Engineering and Computer Sciences, 2005. Ementa de disciplina. Disponível em: <https://people.eecs.berkeley.edu/~klein/cs294-5/index.html>. Acesso em: 20 jul. 2023.

KOJIMA, T. *et al.* Large language models are zero-shot reasoners. *In*: **ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS**, 35., 2022, New Orleans. **Proceedings** [...]. New Orleans: NeurIPS, 2022. p. 22199-22213.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, v. 521, p. 436-444, 2015. DOI:10.1038/nature14539.

LEWIS, M. *et al.* BART: Denoising Sequence-to-Sequence pretraining for natural language generation, translation, and comprehension. **arXiv** [preprint], p. 1-10, 2019. arXiv:1910.13461v1.

LIU, P. *et al.* Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. **ACM Computing Surveys**, v. 55, n. 9, 2023.

LUO, Z. *et al.* WizardCoder: Empowering Code Large Language Models with Evol-Instruct. **arXiv** [preprint], p. 1-26, 20 Jun 2023. arXiv:2306.08568.

MADAAN, A. *et al.* Self-refine: Iterative refinement with self-feedback. **arXiv** [preprint], p. 1-54, 25 May 2023. arXiv:2303.17651.

MARS, M. From Word Embeddings to Pre-Trained Language Models: A State-of-the-Art Walkthrough. **Applied Sciences**, v. 12, n. 17, p. 8805, 2022.

MNIH, V. *et al.* Recurrent models of visual attention. *In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, 27., 2014, Montréal. **Proceedings [...]**, Montréal: NIPS, 2014. p. 1-9.

NOGUEIRA, J. G. ChatGPT bateu 1 milhão de usuários em apenas 5 dias, diz OpenAI. **TecMundo**, [s.l.], 25 jan. 2023, 17:30. Disponível em: <https://www.tecmundo.com.br/software/259681-chatgpt-bateu-1-milhao-usuarios-5-dias-diz-openai.htm>. Acesso em: 20 jul. 2023.

NUNES, D. *et al.* Evaluating GPT-3.5 and GPT-4 Models on Brazilian University Admission Exams. **arXiv** [preprint], p. 1-7, 29 Mar. 2023. arXiv:2303.17003v1.

OPENAI. **Introducing ChatGPT**. 30 Nov. 2022. Disponível em: <https://openai.com/blog/chatgpt>. Acesso em: 20 jul. 2023.

OPENAI. **Model index for researchers**. Copyright 2023. Disponível em: <https://platform.openai.com/docs/model-index-for-researchers>. Acesso em: 20 jul. 2023.

OPENAI. **Models**. Copyright 2023. Disponível em: <https://platform.openai.com/docs/models>. Acesso em: 20 jul. 2023.

OPENAI. GPT-4 technical report. **arXiv** [preprint], 27 Mar. 2023. 100 p. arXiv:2303.08774v3.

PATEL, R.; PATEL, S. Deep learning for natural language processing. *In: KAISER, M. S.; XIE, J.; RATHORE, V. S. (eds.). Information and Communication Technology for Competitive Strategies (ICTCS 2020)*. [S.l.]: Springer Singapore, 2021. p. 523-533. Lecture Notes in Networks and Systems.

PIRES, R. *et al.* Sabiá: Portuguese Large Language Models. **arXiv** [preprint], p. 1-15, 20 July 2023. arXiv:2304.07880.

QIAO, S. *et al.* Reasoning with language model prompting: A survey. **arXiv** [preprint], p. 1-24, 19 Dec 2022. arXiv:2212.09597.

RADFORD, A. *et al.* **Improving language understanding by generative pre-training**. [preprint]. 2018. 12 p.

SCHMIDHUBER, J. Deep learning in neural networks: an overview. **Neural Networks**, v. 61, p. 85-117, 2015. DOI:10.1016/j.neunet.2014.09.003.

SCHUSTER, M.; PALIWAL, K. K. Bidirectional recurrent neural networks. **IEEE Transactions on Signal Processing**, v. 45, n. 11, p. 2673-2681, 1997.

SENGUPTA, S. *et al.* A review of deep learning with special emphasis on architectures, applications and recent trends. **Knowledge-Based Systems**, Amsterdam, v. 194, 2020.

SHRESTHA, A., MAHMOOD, A. Review of Deep Learning Algorithms and Architectures. **IEEE Access**, v. 7, 2019.

SILVEIRA, I. C.; MAUÁ, D. D. Advances in automatically solving the ENEM. *In: BRAZILIAN CONFERENCE ON INTELLIGENT SYSTEMS (BRACIS), 7., 2018, São Paulo. Proceedings [...].* São Paulo: IEEE, 2018. p. 43-48.

SINGH, S.; MAHMOOD, A. The NLP cookbook: modern recipes for transformer based deep learning architectures. **IEEE Access**, v. 9, p. 68675-68702, 2021.

SINGHAL, K. *et al.* Large language models encode clinical knowledge. **Nature**, 620, p. 172-180, 2023.

SUN, C. *et al.* Revisiting unreasonable effectiveness of data in deep learning era. *In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV), 2017, Venice. Proceedings [...].* Venice: IEEE, 2017. p. 843-852.

SUTSKEVER, I.; VINYALS, O.; LE, Q. V. Sequence to sequence learning with neural networks. *In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 27., 2014, Montréal. Proceedings [...],* Montréal: NIPS, 2014.

TERENCE, A. C. F.; ESCRIVÃO FILHO, E. Abordagem quantitativa, qualitativa e a utilização da pesquisa-ação nos estudos organizacionais. *In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 26., 2006, Fortaleza. Anais [...]* Fortaleza: ABEPRO; UFCE, 2006. Disponível em: http://www.abepro.org.br/biblioteca/enegep2006_tr540368_8017.pdf. Acesso em: 24 jul. 2022.

THE GUARDIAN. ChatGPT reaches 100 million users two months after launch. **The Guardian**, London, 02 Feb 2023. Disponível em: <https://www.theguardian.com/technology/2023/feb/02/chatgpt-100-million-users-open-ai-fastest-growing-app>. Acesso em: 20 jul. 2023.

VASWANI, A. *et al.* Attention is all you need. *In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 31., 2017, Long Beach. Proceedings [...],* Long Beach: NIPS, 2017. p. 1-11.

WANG, L. *et al.* Plan-and-solve prompting: improving zero-shot chain-of-thought reasoning by large language models. **arXiv** [preprint], p. 1-24, 6 May 2023. arXiv:2305.04091.

WANG, X. *et al.* Self-consistency improves chain of thought reasoning in language models. **arXiv** [preprint], p. 1-24, 21 Mar. 2022. arXiv:2203.11171.

WEI, J. *et al.* Chain-of-thought prompting elicits reasoning in large language models. *In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 35., 2022a, New Orleans. Proceedings [...].* New Orleans: NeurIPS, 2022. p. 24824-24837.

WEI, J. *et al.* Emergent abilities of Large Language Models. **Transactions on Machine Learning Research**, p. 1-30, Aug. 2022b. Disponível em: <https://openreview.net/pdf?id=yzkSU5zdwD>. Acesso em: 20 abr. 2023.

WOLF, T. *et al.* Transformers: state-of-the-art natural language processing.
In: CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE
PROCESSING: SYSTEM DEMONSTRATIONS, 2020, [virtual]. **Proceedings** [...]. Seattle:
ACL, 2020. p. 38-45.

WU, C. *et al.* Pmc-llama: Further finetuning llama on medical papers. **arXiv** [preprint], p. 1-7, 20 May. 2023. arXiv:2304.13712v2.

YANG, J. *et al.* Harnessing the power of llms in practice: a survey on ChatGPT and beyond. **arXiv** [preprint], p. 1-24, 27 Apr. 2023. arXiv:2304.13712v2.

YANG, Z. *et al.* XLNet: generalized autoregressive pretraining for language understanding.
In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 32., 2019,
Vancouver. **Proceedings** [...]. Vancouver: NeurIPS, 2019.

ZHANG, X. *et al.* Evaluating the Performance of Large Language Models on GAOKAO Benchmark. **arXiv** [preprint], p. 1-4, 23 May 2023. arXiv:2305.12474.

ZHAO, W. X. *et al.* A survey of large language models. **arXiv** [preprint], 29 June 2023. p. 1-85. arXiv:2303.18223v11.

ZHONG, W. *et al.* AGIEval: A human-centric benchmark for evaluating foundation models. **arXiv** [preprint], p. 1-20, 13 Apr. 2023. arXiv:2304.06364.