

Ajuste Fino de Grandes Modelos de Linguagem para Análise de Propriedades Textuais no DrugBank

Felipe Rocha Verol^{1,2}, André Gomes Regino¹, Rodrigo Bonacin¹

f248552@dac.unicamp.br, aregino@cti.gov.br, rodrigo.bonacin@cti.gov.br

¹Divisão de Metodologias da Computação – DIMEC
CTI/MCTI Renato Archer – Campinas/SP

²Instituto de Computação – IC
Universidade Estadual de Campinas – Campinas/SP

Abstract. *This article investigates the efficiency of trained language models compared to untrained models in answering questions about the metabolism and absorption of pharmacological substances, using data from DrugBank. For this purpose, a LLaMA 3.1 8B model was trained with a dataset composed of questions and answers extracted from this information. For evaluation, both the trained model and the base model answered the same 500 questions, and their responses were assessed using the ROUGE-L, BLEU, and Exact Match metrics. The results showed that the base model was outperformed in all three metrics by the trained model, demonstrating that training on specific topics enhances model performance.*

Resumo. *Este artigo investiga a eficiência de modelos de linguagem treinados em relação a modelos não treinados na tarefa de responder perguntas sobre o metabolismo e absorção de substâncias farmacológicas, utilizando dados do DrugBank. Para isso, um modelo LLaMA 3.1 8B foi treinado com uma base de dados formada de perguntas e respostas extraídas dessas informações. Para avaliação, o modelo treinado e o modelo base responderam às mesmas 500 perguntas e suas respostas foram avaliadas por meio das métricas ROUGE-L, BLEU e Exact Match. Os resultados obtidos revelaram que o modelo base foi superado nas três métricas analisadas pelo modelo treinado, demonstrando que o treinamento em assuntos específicos aprimora o desempenho do modelo.*

1. Introdução

Grandes Modelos de Linguagem (LLMs, do inglês *Large Language Models*) têm alcançado avanços significativos no campo do Processamento de Linguagem Natural (PLN), demonstrando alta capacidade em tarefas de tradução, sumarização e resposta a perguntas. Contudo, modelos base, que não são treinados para domínios específicos, tendem a apresentar desempenho inferior quando aplicados a áreas de conhecimento altamente especializadas [ANISUZZAMAN, 2025], como a farmacologia, podendo gerar respostas imprecisas ou inconsistentes em relação ao esperado.

A base de dados do *DrugBank*¹ reúne informações detalhadas e precisas sobre aspectos químicos, biológicos e farmacológicos de diversas substâncias. Entre esses

¹ <https://go.drugbank.com/>

dados, destacam-se as propriedades de metabolismo e absorção. O metabolismo refere-se ao conjunto de reações químicas responsáveis por transformar uma substância em metabólitos, geralmente mais hidrossolúveis, com o objetivo de facilitar sua excreção. Já a propriedade de absorção descreve como, em que quantidade e em quanto tempo uma substância é absorvida pelo organismo humano. Contudo, a consulta e interpretação dessas informações pode ser complexa e trabalhosa, pois se tratam de campos textuais extensos, densos e altamente técnicos.

Nesse contexto, o ajuste fino de LLMs a partir de conjuntos de dados específicos apresenta-se como uma abordagem promissora para aprimorar o desempenho desses modelos em tarefas de resposta a perguntas de áreas de conhecimento complexas. Este trabalho tem como objetivo avaliar o desempenho de um modelo *LLaMA* 3.1 8B treinado com um conjunto de perguntas e respostas sobre metabolismo e absorção de fármacos presentes na base de dados do *DrugBank*, comparando-o ao seu respectivo modelo base, não treinado nesse domínio.

No processo de avaliação, ambos os modelos responderam ao mesmo conjunto de 500 perguntas sobre metabolismo e absorção de fármacos, selecionadas aleatoriamente. As respostas foram posteriormente analisadas por meio de três métricas distintas: *ROUGE-L*, *BLEU* e *Exact Match*. Os resultados obtidos evidenciaram resultados superiores do modelo treinado em relação ao modelo base, como um aumento de 2130% na métrica *BLEU*, reforçando a importância e a eficácia do ajuste fino como ferramenta para aprimorar a qualidade das respostas em domínios especializados.

O restante do artigo está disposto da seguinte forma: a Seção 2 apresenta o referencial teórico, a Seção 3 descreve a metodologia utilizada, a Seção 4 apresenta os resultados obtidos e, finalmente, a Seção 5 discorre sobre as conclusões tiradas.

2. Referencial teórico

Esta seção apresenta os fundamentos teóricos que sustentam a pesquisa, destacando conceitos teóricos que possibilitam uma melhor compreensão das próximas seções dispostas neste artigo. A Seção 2.1 descreve o *DrugBank*; a Seção 2.2 apresenta o conceito de LLMs; a Seção 2.3 descreve o conceito de ajuste fino dos LLMs e a Seção 2.4 descreve os métodos de avaliação dos resultados produzidos pelos LLMs.

2.1. *DrugBank*

O *DrugBank* é uma base de dados biomédica e farmacológica amplamente utilizada por pesquisadores ao redor do mundo [PROBST, 2018; MURUGAN, 2020; RIBEIRO, 2021]. Ela fornece informações confiáveis e precisas sobre diversos fármacos, incluindo suas propriedades de absorção, distribuição, metabolismo, excreção, toxicidade, entre outras. Além disso, a plataforma é constantemente atualizada, garantindo que os pesquisadores tenham acesso a informações atuais e verificadas, tornando-se uma ferramenta importante para estudos na área médica e farmacêutica.

2.2. Grandes Modelos de Linguagem

Grandes Modelos de Linguagem são modelos de aprendizado profundo, geralmente baseados na arquitetura de *transformers* [VASWANI, 2017]. Sua principal característica é a capacidade de compreender, gerar e manipular texto em linguagem natural, o que os torna ferramentas versáteis em diferentes contextos. Modelos como o GPT ganharam notoriedade nos últimos anos, popularizando a tecnologia e consolidando seu uso em tarefas como tradução [CHAN, 2024], sumarização [POKALE, 2023], resposta a perguntas [TAN, 2023] e demais aplicações de Processamento de Linguagem Natural.

O diferencial das LLMs está no volume massivo de dados empregados durante o processo de pré-treinamento, o que lhes confere uma compreensão mais sofisticada de estruturas sintáticas e semânticas. Dessa forma, tais modelos podem atuar como assistentes de propósito geral, adaptando-se a diferentes cenários de uso. Além disso, através de processos de especialização, como o ajuste fino, tornam-se capazes de lidar com domínios específicos, destacando-se como ferramentas eficientes em áreas que exigem precisão, como a biomédica e farmacológica.

2.3. Ajuste fino (*fine-tuning*)

LLMs se destacam por sua alta capacidade de compreender e gerar textos, o que as torna ferramentas poderosas de propósito geral. No entanto, para que possam executar tarefas em domínios específicos, é necessário um processo adicional de especialização. É nesse contexto que se insere o ajuste fino (*fine-tuning*).

O ajuste fino consiste em retreinar um modelo pré-treinado, de modo a ajustar seus parâmetros às exigências de uma tarefa ou domínio específicos. Para isso, utiliza-se um conjunto de dados específico, que orienta o modelo quanto ao comportamento esperado [ANISUZZAMAN, 2025]. Diferentemente do processo de pré-treinamento, esse processo requer uma quantidade consideravelmente menor de dados, já que o modelo já possui a capacidade de compreender e gerar texto, e o ajuste atua apenas para alinhar essa capacidade ao objetivo desejado.

Existem, no entanto, diferentes abordagens para ajuste fino, a depender do propósito do ajuste e das limitações técnicas do ambiente de treinamento:

- ***Fine-tuning completo***: refere-se ao ajuste de todos os parâmetros do modelo base, sendo indicado quando se busca uma modificação ampla em seu comportamento. Apesar de sua eficácia, essa abordagem demanda elevado custo computacional.
- ***Adapter tuning***: engloba métodos que ajustam apenas um subconjunto específico de parâmetros, inseridos de forma estratégica no modelo. Essa técnica possibilita uma adaptação eficiente, preservando a maior parte da estrutura original e reduzindo os custos de treinamento.

- **Quantização:** consiste em representar os parâmetros do modelo em precisões menores (comumente 4 *bits*). Embora acarrete uma pequena perda de precisão, essa estratégia torna o modelo mais leve e eficiente, facilitando sua utilização em cenários com restrições computacionais.

Para esse trabalho, foi empregada a técnica QLoRA [DETTMERS, 2023], que consiste em carregar um modelo em quantização de 4 *bits* e utilizar o método LoRA, uma forma de *adapter tuning* que insere matrizes adicionais de baixa dimensão nas camadas do modelo, permitindo o ajuste de apenas uma pequena fração dos parâmetros.

2.4. Métricas de avaliação de LLMs

Para avaliar as respostas geradas pelos modelos, foram utilizadas as três métricas distintas: *ROUGE-L*, *BLEU* e *Exact Match*. Cada uma delas possui métodos de avaliação distintos, de maneira que as respostas possam ser avaliadas de diferentes perspectivas de similaridade em relação às respostas de referência.

A métrica *ROUGE-L* (*Recall-Oriented Understudy for Gisting Evaluation - Longest Common Subsequence*) se baseia na identificação da maior subsequência comum entre a resposta obtida por um modelo e a resposta esperada, considerando a ordem das palavras. Dessa forma, esta é uma métrica sintática, que avalia o quão próxima a resposta do modelo é da resposta de referência em termos de sequência de *tokens*. Esta é uma técnica muito utilizada para avaliar resumos, pois identifica corretamente se certas palavras chave estão presentes na ordem esperada, e seu resultado é sempre uma nota entre 0 e 1. [LIN, 2004]

Já a métrica *BLEU* (*Bilingual Evaluation Understudy*), que foi desenvolvida inicialmente para avaliação de traduções, avalia a sobreposição de *n*-gramas entre a resposta gerada pelo modelo e a resposta esperada. Essa métrica identifica e penaliza divergências na escolha lexical, sendo uma métrica de caráter sintático. Assim como a métrica *ROUGE-L*, as notas retornadas também variam de 0 a 1. [PAPINENI, 2002]

A métrica *Exact Match* avalia de forma simples e binária: caso a resposta obtida pelo modelo seja exatamente igual a resposta de referência, retorna 1. Caso contrário, retorna 0. Essa é uma boa estratégia para avaliar se o modelo consegue reproduzir fielmente as respostas esperadas, servindo como um indicador estrito de exatidão. [TERVEN, 2023]

O conjunto de avaliações gerado por essas três métricas oferece uma visão equilibrada: enquanto *ROUGE-L* e *BLEU* capturam um grau de similaridade lexical e estrutural, o *Exact Match* avalia a fidelidade absoluta das respostas obtidas.

3. Metodologia

A execução dos experimentos teve como ferramenta central a técnica de ajuste fino (*fine-tuning*), aplicada ao modelo *LLaMA* 3.1 8B. Antes, contudo, foi necessário realizar um processo de tratamento e preparação dos dados provenientes do *DrugBank*, descrito na

subseção seguinte. Todos os *scripts* e arquivos citados nesta seção estão no repositório do *GitHub*².

3.1. Preparação dos dados

Inicialmente, foi feito *download* do site do *DrugBank* um arquivo no formato XML de 1.58 GB, contendo os campos textuais de 2628 drogas da base de dados. Para este estudo, foi necessário extrair especificamente as informações relativas às propriedades de metabolismo e de absorção de cada substância. Para tanto, desenvolveu-se um script denominado *extraí-metabolism-absorption.py*, executado no ambiente *do Kaggle devido ao elevado tamanho do arquivo original*, o que inviabiliza o processamento local. A execução do script resultou em um novo arquivo, denominado *metabolism_absorption.xml*, estruturado em tags `<drug>`, contendo, em seu interior, os campos `<name>` (nome da droga), `<metabolism>` (informações textuais sobre o metabolismo da substância) e `<absorption>` (informações textuais sobre a absorção da substância), conforme apresentado na Figura 1.

```
<drug type="small molecule" created="2005-06-13" updated="2023-01-03">
  <name>Bivalirudin</name>
  <metabolism>80% proteolytic cleavage</metabolism>
  <absorption>Following intravenous administration, bivalirudin exhibits linear pharmacokinetics.
    The mean steady state concentration is 12.3 +/- 1.7mcg/mL after administration of an
    intravenous bolus of 1mg/kg followed by a 2.5mg/kg/hr intravenous infusion given over 4 hours.</absorption>
</drug>
```

Figura 1. Exemplo de estrutura do arquivo *metabolism_absorption.xml*

A seguir, foi necessário formatar esses dados para que fossem utilizados como pares pergunta/resposta no processo de treinamento. Para isso, foi utilizado o script *metabolism-absorption-prompts.py*, que é responsável por abrir e processar os arquivos *metabolism_absorption.xml* e *negative_examples.txt*. Este último contém exemplos de perguntas de outros domínios de conhecimento geral, acompanhadas sempre da mesma resposta padronizada, indicando que o modelo não pode responder sobre aquele assunto, auxiliando no ajuste do comportamento do modelo, de modo que ele se restrinja a responder apenas perguntas relacionadas ao metabolismo e à absorção de substâncias. O *script* cria quatro tipos de perguntas sobre o metabolismo e quatro tipos de perguntas sobre a absorção de cada droga. Além disso, utiliza também as perguntas provenientes do arquivo *negative_examples.txt*. Todos esses exemplos são formatados em JSON, onde cada item contém as chaves “*instruction*”, que armazena a pergunta em si, “*input*”, que contém o nome da droga à qual a pergunta se refere, e “*output*”, que contém a resposta esperada, obtida a partir dos campos textuais previamente extraídos do *DrugBank*. Por fim, um arquivo JSON, exemplificado na Figura 2, denominado *metabolism_absorption_alpaca.json* é gerado, contendo o necessário para se iniciar o processo de treinamento. Este arquivo encontra-se armazenado em um repositório Hugging Face³.

² https://github.com/felipeverol/drugbank_finetuning

³ https://huggingface.co/datasets/verolfelipe/drug_bank_metabolism_absorption_alpaca

```
{
  "instruction": "Can you explain how Oprelvekin is absorbed in the body?",
  "input": "Oprelvekin",
  "output": "Absolute bioavailability is over 80%."
},
{
  "instruction": "Describe the absorption characteristics of Oprelvekin.",
  "input": "Oprelvekin",
  "output": "Absolute bioavailability is over 80%."
},
{
  "instruction": "Tell me about Oprelvekin's absorption mechanism.",
  "input": "Oprelvekin",
  "output": "Absolute bioavailability is over 80%."
},
{
  "instruction": "In what way is Oprelvekin absorbed?",
  "input": "Oprelvekin",
  "output": "Absolute bioavailability is over 80%."
},
}
```

Figura 2. Exemplo de estrutura do arquivo *metabolism_absorption_alpaca.json*

3.2. Treinamento

Para o treinamento do modelo, foram utilizadas as bibliotecas *Unsloth*⁴, *TRL*⁵ e *Datasets*⁶, que simplificam e otimizam o processo de *fine-tuning* em grandes modelos de linguagem. O script de treinamento, denominado *llama-3.1-8b-unsloth.py*, foi desenvolvido com base no *notebook* oficial disponibilizado pela documentação da *Unsloth* para ajuste fino do modelo *LLaMA 3.1 8B*. O processo emprega técnicas de quantização em 4 *bits* (QLoRA) e LoRA (*Low-Rank Adaptation*), que permitem reduzir os requisitos computacionais mantendo a capacidade de adaptação do modelo. Para sua execução, foi utilizada uma GPU *NVIDIA RTX A4000* com 16 GB de VRAM.

Inicialmente, o modelo *LLaMA 3.1 8B* foi carregado já quantizado em 4 *bits* a partir do repositório *unsloth/Meta-Llama-3.1-8B* disponível no *Hugging Face*. Em seguida, foram realizadas as configurações para o LoRA, adotando como hiperparâmetros principais: $r = 16$, $lora_alpha = 16$, $lora_dropout = 0$ e $target_modules = ["q_proj", "k_proj", "v_proj", "o_proj", "gate_proj", "up_proj", "down_proj"]$.

Posteriormente, o *dataset* foi carregado por meio da biblioteca *Datasets*, sendo formatado no padrão *Alpaca Prompt Template*. Essa estrutura apresenta um formato claro de perguntas e respostas, o que assegura a consistência no processamento dos *prompts* e orienta o modelo base a compreender de que maneira deve responder.

As configurações de treinamento foram definidas utilizando os objetos das classes *SFTTrainer* e *SFTConfig*, fornecidos pela biblioteca *TRL*. Entre os principais hiperparâmetros adotados, destacam-se: $num_train_epochs = 2$, $per_device_train_batch_size = 2$, $gradient_accumulation_steps = 4$ e $learning_rate = 2e-4$. O treinamento teve duração aproximada de 6 horas.

⁴ <https://unsloth.ai/>

⁵ <https://huggingface.co/docs/trl/index>

⁶ <https://huggingface.co/docs/datasets/index>

Ao final do processo, tanto os *adapters* LoRA quanto a versão *mergeada* em 16 *bits* do modelo, que consiste nos *adapters* somados ao modelo base, foram enviados para repositórios no *Hugging Face*, viabilizando sua utilização em inferências e experimentos posteriores.

3.3. Avaliação dos modelos

Para a avaliação quantitativa do modelo, foram utilizados dois *scripts* distintos. O primeiro, denominado *gera_respostas_csv.py*, tem como função principal instanciar tanto o modelo treinado quanto o modelo base. Esse *script*, executado no mesmo ambiente em que o modelo foi treinado, seleciona aleatoriamente 500 perguntas do *dataset* previamente utilizado no processo de *fine-tuning* e realiza uma inferência para cada pergunta em ambos os modelos. As respostas geradas são armazenadas no arquivo *avaliacao_500.csv*, que contém as colunas *pergunta*, *resposta_esperada*, *resposta_modelo_treinado* e *resposta_modelo_base*. Este conjunto é usado para verificar a capacidade de recuperação e de apresentação em linguagem natural dos dados nele contidos, por meio de técnicas de ajuste fino do modelo. Estando fora do escopo a avaliação do comportamento do modelo em um novo *dataset*.

O segundo *script*, denominado *gera_metricas_csv.py*, é responsável por avaliar as respostas de ambos os modelos com base em três métricas distintas: *ROUGE-L*, *BLEU* e *Exact Match*. Para cada resposta, o *script* adiciona ao arquivo *avaliacao_500.csv* seis novas colunas: *rougeL_base*, *bleu_base*, *exact_base*, *rougeL_ft*, *bleu_ft* e *exact_ft*, correspondentes às pontuações atribuídas às respostas do modelo base e do modelo treinado, respectivamente. Ao final da execução, o *script* imprime as médias de cada métrica para cada modelo, evidenciando, de forma quantitativa, o desempenho médio relativo entre o modelo ajustado e o modelo original.

Para uma avaliação complementar, de caráter mais informal e qualitativo, utilizou-se o *script* *testar_llama.py*. Esse recurso permite que o usuário formule perguntas diretamente a ambos os modelos e compare suas respostas. Tal abordagem não substitui a análise quantitativa, mas oferece exemplos ilustrativos que ajudam a explicitar, de maneira mais concreta, os resultados obtidos.

4. Resultados

Esta seção dedica-se a discutir os resultados obtidos a partir das avaliações sob duas perspectivas distintas: quantitativa e qualitativa. A primeira busca mensurar a qualidade das respostas por meio de métricas numéricas, enquanto a segunda permite visualizar e comparar as respostas por meio de inferência.

4.1. Resultados quantitativos

Os resultados da avaliação quantitativa estão apresentados na tabela a seguir, a qual reúne as médias das métricas consideradas de cada modelo: *ROUGE-L*, *BLEU* e *Exact Match*.

	Modelo base	Modelo ajustado
ROUGE-L	0.1129	0.3815
BLEU	0.0097	0.2164
Exact Match	0.0000	0.1460

Tabela 1. Métricas obtidas pelo modelo base e pelo modelo ajustado

A partir dos resultados obtidos, observa-se uma evolução consistente em todas as métricas avaliadas. Comparando, o modelo ajustado apresentou um aumento de aproximadamente 238% em *ROUGE-L* e 2130% em *BLEU* em relação ao modelo base. Além disso, a métrica *Exact Match* passou de 0 para 0.1460, o que indica que o modelo, antes incapaz de reproduzir qualquer resposta de forma fidedigna, passou a apresentar um desempenho significativamente melhor. Esse avanço evidencia que o modelo treinado apresentou maior aderência às respostas de referência e também produziu saídas mais próximas ao padrão esperado. Em outras palavras, os resultados quantitativos indicam melhora na capacidade do modelo em compreender e reproduzir aquilo que é esperado, sugerindo que o processo de *fine-tuning* contribuiu de maneira efetiva para o aprimoramento do desempenho.

4.2. Resultados qualitativos

Para tornar os resultados mais evidentes, apresenta-se a seguir um exemplo qualitativo obtido por meio da execução do *script testar_llama.py*:

```
Enter the instruction: Can you explain how Oprelvekin is absorbed in the body?

--- LoRA Model Response ---
<|begin_of_text|>Below is an instruction that describes a task. Write a response that appropriately completes the request.

### Instruction:
Can you explain how Oprelvekin is absorbed in the body?

### Response:
Absolute bioavailability is 80% following subcutaneous administration.<|end_of_text|>

--- Base Model Response ---
<|begin_of_text|>Below is an instruction that describes a task. Write a response that appropriately completes the request.

### Instruction:
Can you explain how Oprelvekin is absorbed in the body?

### Response:
Oprelvekin is absorbed in the body by the intestinal epithelial cells, which are specialized cells that line the intestines. These cells have a specific structure that allows them to absorb nutrients from the food we eat and transport them to the bloodstream. When Oprelvekin is taken orally, it is absorbed through the intestinal epithelial cells and transported to the bloodstream, where it can be used by the body to treat various medical conditions.

### Instruction:
Can you explain how Oprelvekin is absorbed in the body?

### Response:
Oprelvekin is absorbed in the body through the gastrointestinal tract, which is
```

Figura 3. Exemplo de inferência ao modelo base e ao modelo ajustado

Considerando que a resposta esperada, segundo o dataset utilizado para o treinamento, era “*Absolute bioavailability is over 80%*”, dois pontos principais podem ser destacados. Em primeiro lugar, embora o modelo ajustado não tenha apresentado um desempenho totalmente ideal, já que acrescentou a informação de que o fármaco é

administrado por via subcutânea, sua resposta foi significativamente mais próxima do esperado, pois conseguiu reproduzir corretamente o dado central de que a biodisponibilidade absoluta é de 80%. Em segundo lugar, observa-se que o modelo ajustado demonstrou compreender de maneira mais precisa o formato de interação de pergunta e resposta esperado, limitando-se a fornecer a resposta solicitada. O modelo base, por outro lado, além de responder à questão inicial, elaborou uma nova pergunta para si mesmo e a respondeu em seguida, comportamento que evidencia uma falha em reconhecer sua função de responder apenas ao que foi perguntado.

5. Conclusão

Os resultados obtidos evidenciam que o ajuste fino do modelo *LLaMA* 3.1 8B resultou em melhorias notáveis de desempenho, tanto em métricas quantitativas quanto em análises qualitativas. O modelo ajustado mostrou-se mais alinhado ao conteúdo semântico do *dataset*, gerando respostas com informações relevantes e no formato adequado. Além disso, destacou-se pela capacidade de manter o foco no escopo definido, respondendo exclusivamente a questões relacionadas ao metabolismo e à absorção de fármacos.

Portanto, os resultados reforçam a importância do fine-tuning como método de especialização de Grandes Modelos de Linguagem, garantindo maior confiabilidade e aplicabilidade para escopos específicos. No caso estudado, o aumento da capacidade do modelo treinado demonstra que LLMs podem ser usadas como uma poderosa ferramenta de apoio à pesquisa científica.

Referências

- ANISUZZAMAN, D. M. et al. Fine-tuning large language models for specialized use cases. *Mayo Clinic Proceedings: Digital Health*, v. 3, n. 1, p. 100184, 2025.
- CHAN, Venus; TANG, William Ko-Wai. GPT for translation: A systematic literature review. *SN computer science*, v. 5, n. 8, p. 986, 2024.
- DETTMERS, Tim et al. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, v. 36, p. 10088-10115, 2023.
- LIN, Chin-Yew. Rouge: A package for automatic evaluation of summaries. In: *Text summarization branches out*. 2004. p. 74-81.
- MURUGAN, Natarajan Arul et al. Searching for target-specific and multi-targeting organics for Covid-19 in the Drugbank database with a double scoring approach. *Scientific reports*, v. 10, n. 1, p. 19125, 2020.
- PAPINENI, Kishore et al. Bleu: a method for automatic evaluation of machine translation. In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002. p. 311-318.
- POKALE, Sairaj et al. Text summarization: GPT perspective. In: *2023 3rd Asian Conference on Innovation in Technology (ASIANCON)*. IEEE, 2023. p. 1-7.

PROBST, Daniel; REYMOND, Jean-Louis. Exploring DrugBank in virtual reality chemical space. *Journal of chemical information and modeling*, v. 58, n. 9, p. 1731-1735, 2018.

RIBEIRO, Hugo et al. Opioides em ambulatório na dor não oncológica: uma revisão sobre os desafios da farmacologia no envelhecimento. *Revista Portuguesa de Medicina Geral e Familiar*, v. 37, n. 3, p. 233-41, 2021.

TAN, Yiming et al. Can ChatGPT replace traditional KBQA models? An in-depth analysis of the question answering performance of the GPT LLM family. In: *International Semantic Web Conference*. Cham: Springer Nature Switzerland, 2023. p. 348-367.

TERVEN, Juan et al. Loss functions and metrics in deep learning. *arXiv preprint arXiv:2307.02694*, 2023.

VASWANI, Ashish et al. Attention is all you need. *Advances in neural information processing systems*, v. 30, 2017.