Análise Quantitativa do Trade-off entre Especialização e Generalização em LLMs via Fine-Tuning para Text-to-SQL

Santos. Jacó, *Discente, Universidade Federal do Amazonas (UFAM), Instituto de Computação (IComp)* Curso**:** *ICC220 - Tópicos Especiais em Bancos de Dados / PPGINF528 - Tópicos Especiais em Recuperação de Informação*
Professores: Dr. André de Carvalho, Dr. Altigran da Silva

Resume—Este trabalho conduz uma avaliação empírica do processo de fine-tuning em Modelos de Linguagem de Grande Porte (LLMs) para a tarefa especializada de Text-to-SQL. Utilizamos o modelo mistralai/Mistral-7B-Instruct-v0.2 como base e o dataset Spider para treinar duas variantes com diferentes hiperparâmetros. O desempenho foi medido com uma métrica customizada de Acurácia de Execução, que avalia a correção funcional das queries SQL geradas. Os resultados indicam um ganho de especialização de até 144% na tarefa-alvo, mas com uma completa degradação (esquecimento catastrófico) da capacidade de conhecimento geral, medida pelo benchmark MMLU. Concluímos que, embora o fine-tuning seja altamente eficaz para criar modelos especialistas, ele o faz ao custo de suas habilidades generalistas, uma consideração crítica para o desenvolvimento de aplicações comerciais.

Index Terms— Large Language Models, Fine-Tuning, Text-to-SQL, Semantic Parsing, Catastrophic Forgetting, Low-Rank Adaptation (LoRA), Performance Evaluation.

1. Introdução

Modelo de Linguagem de Grande Porte (LLMs) emergiram como tecnologias transformadoras, demonstrando capacidades notáveis em uma vasta gama de tarefas de processamento de linguagem natural. Uma técnica proeminente para otimizar seu desempenho em domínios específicos é o fine-tuning. Este processo adapta um modelo pré-treinado a um novo conjunto de dados, especializando-o na tarefa desejada, como a tradução de linguagem natural para consultas SQL (Text-to-SQL).

Contudo, a especialização intensiva pode levar a um fenômeno adverso conhecido como "esquecimento catastrófico" (catastrophic forgetting), onde o modelo perde sua capacidade de realizar tarefas fora do domínio de fine-tuning. Este trabalho apresenta uma avaliação empírica e sistemática deste trade-off. O objetivo central é projetar e executar um pipeline experimental para quantificar o ganho de desempenho na tarefa de Text-to-SQL e, simultaneamente, medir a perda de robustez em domínios de conhecimento geral [2].

2. Metodologia

Para garantir a reprodutibilidade dos experimentos, todas as operações estocásticas foram controladas com sementes de aleatoriedade fixas (seed=42). O ambiente computacional utilizado foi o Google Colab com uma GPU T4 com armazenamento de pastas de grandes volumes no *Google Drive*.

2.1. Configuração Experimental

- Modelo Base: Foi utilizado o modelo *open-source mistralai/Mistral-7B-Instruct-v0.2*, um LLM da classe de 7 bilhões de parâmetros. O modelo foi carregado em 4-

bit utilizando a biblioteca *bitsandbytes* para otimizar o uso de memória.

- *Dataset* de *Fine-Tuning*: Para o *fine-tuning*, utilizou-se exclusivamente o *training split* do *Spider Dataset*, um *benchmark* padrão para *Text-to-SQL*.
- Dataset de Avaliação de Tarefa: A avaliação de desempenho na tarefa-alvo foi realizada utilizando uma amostra de 100 exemplos do *development split* do *Spider Dataset*, uma decisão pragmática devido a restrições de tempo e recursos computacionais. [1]

- Dataset de Avaliação de Generalização: Para medir a regressão de capacidade, foi criada uma suíte de avaliação com 150 questões do benchmark MMLU (*Massive Multitask Language Understanding*), divididas igualmente em três subcategorias:

    - 50 questões de *high school computer* science (*STEM*).
    - 50 questões de *philosophy* (Humanidades).
    - 50 questões de *high school macroeconomics* (Ciências Sociais)

2.2. Processo de *Fine-Tuning*

O *fine-tuning* foi implementado com a técnica de Adaptação de Baixa Ordem (LoRA), uma forma de *Parameter-Efficient Fine-Tuning* (PEFT), utilizando as bibliotecas t*ransformers* e trl do *Hugging Face*.

Fine-tuning é o processo de continuar o treinamento de um modelo já pré-treinado em um novo *dataset*, geralmente menor e mais específico, para adaptá-lo a uma tarefa particular. Para mitigar o alto custo computacional do fine-tuning completo de LLMs, utilizam-se técnicas de PEFT, que atualizam apenas uma pequena fração dos parâmetros do modelo. A técnica LoRA, empregada neste trabalho, congela os pesos originais do LLM e injeta pares de matrizes de baixo ranque (*low-rank*) treináveis em suas camadas, reduzindo drasticamente o número de parâmetros a serem ajustados. A implementação foi realizada com o ecossistema *Hugging Face*, onde a biblioteca *transformers* é responsável por carregar o modelo base, a peft fornece a implementação da adaptação LoRA, e a trl (*Transformer Reinforcement Learning*) oferece o SFTTrainer (*Supervised Fine-tuning Trainer*), um utilitário de alto nível que simplifica o ciclo de treinamento supervisionado.

Foram conduzidos dois experimentos (Run 1 e Run 2) para avaliar o impacto da taxa de aprendizado. A Tabela 1 detalha os hiperparâmetros utilizados.

Tabela 1: Configuração dos Hiperparâmetros de Fine-Tuning

Hiperparâmetro	Configuração Run 1	Configuração Run 2
r (rank)	8	8
lora_alpha	16	16
lora_dropout	0.05	0.05
target_modules	q_proj, v_proj, k_proj, o_proj	q_proj, v_proj, k_proj, o_proj
learning_rate	2e-4	5e-5
num_train_epochs	1	1
optimizer	paged_adamw_8bit	paged_adamw_8bit

2.3. Métrica de Avaliação Customizada: Execution Accuracy

Para uma avaliação fidedigna, foi desenvolvida uma métrica customizada em *DeepEval*, denominada *ExecutionAccuracy*. A arquitetura de software da métrica segue os requisitos do projeto:

1. A classe herda de *deepeval.metrics.BaseMetric*.
2. O método *measure(self, test_case*) se conecta ao banco de dados sqlite correspondente à questão.
3. Executa, de forma segura e dentro de blocos *try-except*, a consulta SQL gerada pelo modelo *(actual_output)* e a consulta gabarito (expected_output).
4. Compara os conjuntos de resultados de forma insensível à ordem das linhas.
5. Retorna 1.0 para sucesso (resultados idênticos) e 0.0 para falha.

3. Resultados

Os resultados foram coletados após a execução dos *pipelines* de avaliação para os três modelos: *Baseline*, *Fine-Tuned Run 1* (usando o *checkpoint-200*), e *Fine-Tuned Run 2*.

3.1. Desempenho na Tarefa-Alvo (*Text-to-SQL*)

A Tabela 2 apresenta a Acurácia de Execução. Os modelos *fine-tuned* demonstraram um ganho de performance substancial sobre o modelo base.

Tabela 2: Resultados da Acurácia de Execução (Spider dev set, n=100)

Modelo	Execution Accuracy
Baseline Model	9.00%
Fine-Tuned Run 1	21.00%
Fine-Tuned Run 2	22.00%

3.2. Regressão de Capacidade (Conhecimento Geral)

A Tabela 3 mostra a acurácia na suíte MMLU. Os resultados indicam um esquecimento catastrófico completo da já limitada capacidade de conhecimento geral do modelo.

Tabela 3: Resultados da Acurácia (MMLU, n=150)

Modelo	STEM	Humanidades	Ciências Sociais	Acurácia Agregada	Variação Agregada
Baseline	0.0%	0.0%	2.0%	0.67%	-
Run 1	0.0%	0.0%	0.0%	0.00%	-100%
Run 2	0.0%	0.0%	0.0%	0.00%	-100%

3.3. Análise de Erros

Uma análise qualitativa das falhas do modelo Run 2 foi realizada. Um erro comum foi a geração de queries sintaticamente válidas, mas com lógica incorreta, especialmente em junções complexas. Por exemplo, para a pergunta *"What are the names, countries, and ages for every singer in*

descending order of age?", o modelo gerou *SELECT T1.Name, T1.Country, T2.Age FROM singer AS T1 JOIN band AS T2 ON T1.Band_ID = T2.Band_ID ORDER BY T2.Age DESC*, realizando uma junção desnecessária com a tabela band e buscando a idade (Age) na tabela errada. A query correta era uma simples consulta na tabela singer: SELECT name, country, age FROM singer ORDER BY age DESC. Este tipo de erro evidencia que, embora o modelo tenha aprendido a estrutura da linguagem SQL, sua compreensão semântica das relações entre tabelas ainda é imperfeita.

4. Discussão

A análise dos resultados permite responder às questões centrais do projeto.

O *trade-off* entre especialização e generalização mostrou-se extremo. Observou-se um ganho de até 13 pontos percentuais na *Execution Accuracy* (um aumento relativo de 144% no Run 2), enquanto a já baixa capacidade de conhecimento geral foi completamente suprimida. Para uma aplicação dedicada, onde a precisão na tarefa-alvo é primordial, este *trade-off* é não apenas justificável, mas altamente desejável.

O principal fator a influenciar este *trade-off* foi a taxa de aprendizado. A *learning_rate* mais baixa do *Run 2* (5e-5) resultou em um modelo marginalmente superior na tarefa-alvo. Notavelmente, o *Run* 1, com uma taxa mais alta (2e-4), demonstrou sinais de colapso em checkpoints posteriores (após o checkpoint-200), embora seu checkpoint inicial tenha apresentado um bom desempenho. Isso reforça a sensibilidade do processo a este hiperparâmetro.

As implicações práticas destes achados são significativas. A especialização de LLMs via *fine-tuning* é uma ferramenta poderosa para criar "especialistas de nicho" eficientes, mas ao custo de suas habilidades generalistas. Para o desenvolvimento de sistemas comerciais, isso sugere que uma arquitetura de múltiplos modelos (um LLM generalista para interação com o usuário e modelos especialistas para tarefas específicas) pode ser mais robusta e eficaz do que tentar criar um único modelo que sirva a todos os propósitos após o fine-tuning.

5. Conclusão

Este trabalho demonstrou quantitativamente o trade-off entre especialização e generalização no fine-tuning de LLMs para a tarefa de *Text-to-SQL*. Comprovamos que é possível mais que dobrar a performance na tarefa-alvo utilizando a técnica LoRA. No entanto, esta especialização ocorre ao custo da completa aniquilação de habilidades, mesmo que residuais, em domínios de conhecimento geral. Os resultados sublinham a importância de uma avaliação multifacetada e uma escolha cuidadosa de hiperparâmetros ao desenvolver LLMs para aplicações específicas. Trabalhos futuros poderiam explorar técnicas de fine-tuning que visam mitigar o esquecimento catastrófico, buscando um equilíbrio mais tênue entre as duas facetas da capacidade de um modelo.

Referências

[1] Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I., Yao, Q., Roman, S., Zhang, Z., & Radev, D. (2018). *Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task*. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.

[2] Carvalho, A., & Silva, A. (2025). Quarto Trabalho Prático: Análise Quantitativa do Trade-off entre Especialização e Generalização em LLMs via Fine-Tuning. Documento da Disciplina ICC220/PPGINF528, Universidade Federal do Amazonas.