Departamento de Ciência da Computação
Arthur Pontes Nader

Fine-Tuning de Modelos de Linguagem para Coleta e Extração de Dados

Orientador: Rodrygo Luis Teodoro Santos

Belo Horizonte
11 de setembro de 2023

1) Introdução

No Projeto Orientado em Computação 1 (POC1), analisou-se diversas ferramentas capazes de facilitar a recuperação de informação por meio do uso de modelos generativos. Essas ferramentas, apesar de representarem um salto importante na área, apresentaram uma série de limitações.

O treinamento desses modelos generativos é uma tarefa que exige recursos computacionais significativos e um conjunto de dados adequado para garantir que o modelo seja capaz de gerar informações relevantes e precisas. Com isso, a técnica de fine-tuning se mostra como uma alternativa valiosa e eficaz para adaptar modelos pré-treinados a tarefas específicas.

Nesse sentido, é útil estudar bibliotecas para processamento de linguagem natural (NLP), como a biblioteca Transformers da Hugging Face, ou frameworks de deep learning, como TensorFlow e PyTorch. Essas ferramentas fornecem implementações eficientes de modelos generativos pré-treinados, como GPT-3, BERT, entre outros, que podem ser ajustados para atender às necessidades específicas de recuperação de informação.

Outro framework que tem ganhado bastante destaque ultimamente é o LangChain, que permite o desenvolvimento de aplicativos alimentados por modelos de linguagem, com foco na capacidade de integração com fontes de dados e na interação com o ambiente. Tem-se também o Llama 2, uma família de modelos de linguagem de grande escala lançados pela Meta, com destaque para modelos de fine-tuning.

Assim, o principal objetivo desse trabalho é explorar essas bibliotecas e frameworks para determinar qual seria um modelo ou arquitetura mais adequado para os diversos tipos de tarefas em recuperação de informação explorados no POC1. Esses conhecimentos serão bastante úteis e podem servir de base para uma pesquisa de mestrado em computação.

2) Referencial Teórico

A arquitetura Transformer, exposta em (VASWANI, et al., 2017), tem sido muito utilizada em modelos para processamento de linguagem natural, sendo inclusive citada em (RADFORD, et al., 2018), um dos artigos iniciais que foram a base para o surgimento dos modelos GPT. Esses dois artigos merecem um destaque quando se trata do entendimento desses modelos de inteligência artificial.

Por sua vez, em (WOLF et al., 2020) é apresentado a biblioteca "transformers", que disponibiliza esses avanços na área para a comunidade de aprendizado de máquina.

Já em (TOUVRON et al., 2023) é mostrado uma coleção de modelos de linguagem de base com até 65 bilhões de parâmetros. Esses modelos são treinados em um volume massivo de dados e destacam a possibilidade de treinar modelos de alta qualidade usando apenas conjuntos de dados publicamente disponíveis, sem depender de dados proprietários.

Por fim, no âmbito de eficiência computacional, é exposto em (GEIPING, GOLDSTEIN, 2022) a possibilidade de treinar um modelo de linguagem em uma única GPU em um dia, propondo modificações para alcançar desempenho comparável a modelos escalados, o que é interessante de se pensar tendo em vista a limitação de recursos computacionais.

3) Metodologia

Inicialmente, é interessante o estudo e a revisão de conceitos fundamentais de modelos generativos, como modelos de linguagem, redes neurais, arquiteturas pré-treinadas e fine-tuning.

Já para exploração e aprendizado das bibliotecas e frameworks citados, há diversos tutoriais disponíveis, além das documentações presentes nos próprios sites ou repositórios.

Outras técnicas de otimização também devem ser exploradas, como por exemplo, a biblioteca PEFT (Parameter-Efficient Fine-Tuning), projetada para permitir a adaptação eficiente de modelos de linguagem pré-treinados a diferentes aplicações sem a necessidade de ajustar todos os parâmetros do modelo.

Em seguida, deve-se avançar para a fase de experimentação, onde começará o ajuste dos modelos e suas aplicações a tarefas específicas. Entre essas tarefas, algumas interessantes seriam o mapeamento de HTML da página para arquivo JSON com a configuração da coleta de dados associada, tradução de comando em linguagem natural para consulta em SQL e raspagem de dados específicos de uma página, tal qual a ferramenta Scrapeghost fazia na POC1.

4) Resultados Esperados

Espera-se que ao final das atividades, tenha-se obtido um conhecimento mais profundo sobre a ampla complexidade que se trata os modelos generativos e os modelos de linguagem.

Além disso, durante o curso do projeto, espera-se adquirir habilidades de aprendizado e implementação de bibliotecas e frameworks específicos, como a biblioteca transformers. O entendimento aprofundado dessas bibliotecas permitirá uma exploração mais eficaz e uma utilização mais eficiente dos recursos disponíveis.

O entendimento das ferramentas LangChain e Llama 2 também abre portas para a exploração de soluções ainda mais avançadas e adaptáveis no campo da recuperação de informações alimentada por modelos de linguagem.

Ao final do projeto, deve ficar mais claro o modo como realizar o fine-tuning desses modelos generativos para as tarefas de recuperação de informação. Dessa forma, ficará mais factível no futuro uma maneira de contornar algumas das limitações das ferramentas apresentadas na POC1.

5) Etapas e Cronogramas

As etapas e cronogramas para desenvolvimento das atividades foram divididas semanalmente da seguinte forma:

Semana	Atividades
1	Revisão de bibliografia
2	Estudo das arquiteturas
3	Estudo da biblioteca transformers, pytorch e tensorflow
4	Análise de códigos envolvendo as bibliotecas
5	Preparação para apresentação parcial
6	Exploração do LangChain
7	Exploração do Llama 2
8	Implementação
9	Implementação
10	Implementação
11	Preparação para apresentação final
12	Escrita do relatório final

6) Referências Bibliográficas

R. Baeza-Yates, B. Ribeiro-Neto. Modern Information Retrieval. 2011. 2nd ed. New York. Addison-Wesley.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, pages 6000–6010, 2017.

A. Radford, K. Narasimhan, T. Salimans, I. Sutskever. Improving Language Understanding by Generative Pre-Training. OpenAI, 2019. Disponível em: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... Rush, A. M. (2020). HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv preprint arXiv:1910.03771v5.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., ... Lample, G. (2023). LLaMA: Open and Efficient Foundation Language Models. arXiv preprint arXiv:2302.13971v1.

Geiping, J., & Goldstein, T. (2022). Cramming: Training a Language Model on a Single GPU in One Day. arXiv preprint arXiv:2212.14034v1.