MINISTRY OF EDUCATION AND TRAINING

**FPT UNIVERSITY**

DeepSeek-based ChatBot system supports work management

by

Dinh Cong Bang

A thesis submitted in conformity with the requirements  
for the degree of Master of Software Engineering

© Copyright by Dinh Cong Bang 2024

MINISTRY OF EDUCATION AND TRAINING

**FPT UNIVERSITY**

DeepSeek-based ChatBot system supports work management

by

Dinh Cong Bang

A thesis submitted in conformity with the requirements  
for the degree of Master of Software Engineering

Supervisor:

1. Assoc. Prof. Phan Duy Hung
2. Dr. Vu Thu Diep

© Copyright by Dinh Cong Bang 2024

DeepSeek-based ChatBot system supports work management

Dinh Cong Bang

Degree Master of Software Engineering

FPT University

2024

Abstract

Trong bối cảnh chuyển đổi số hiện nay, việc kết hợp AI vào các hệ thống là một nhu cầu tất yếu. do đó luận văn này phát triển một hệ thống quản lý công việc nhằm tự động hóa việc tạo báo cáo, hướng dẫn người mới, tra cứu đầu mối. Hệ thống nâng cao năng suất thông qua việc tự động tạo báo cáo, đề xuất nhiệm vụ và hướng dẫn nhân viên mới. Mô hình AI được huấn luyện trên dữ liệu từ Jira và dữ liệu các nhóm chat nội bộ, giúp hệ thống nắm bắt ngữ cảnh thực tế và quy trình nghiệp vụ.

Cốt lõi của hệ thống là mô hình DeepSeek, được tinh chỉnh bằng kỹ thuật LoRA (Low-Rank Adaptation) kết hợp Multi-Stage Fine-Tuning. Kỹ thuật LoRA cố định trọng số gốc và chỉ huấn luyện thêm một lượng nhỏ trọng số hạng thấp, giúp giảm đến hàng nghìn lần số tham số cần cập nhật so với tinh chỉnh toàn bộ mô hình (1). Phương pháp này bảo toàn kiến thức nền nhằm giảm thiểu hiện tượng “quên lệch” (catastrophic forgetting) (2).Ngoài ra, tinh chỉnh đa giai đoạn giúp mô hình duy trì khả năng tổng quát trên nhiều nhiệm vụ (3).Nhờ đó, hệ thống cân bằng tính chuyên môn hóa và khả năng tổng quát, mang lại giải pháp quản lý công việc hiệu quả.

Acknowledgments

Tôi xin bày tỏ lòng biết ơn sâu sắc tới PGS. TS. Phan Duy Hùng – người đã hướng dẫn tôi từ những bước đầu trong việc định hình đề tài và định hướng nghiên cứu. Công trình này sẽ không thể hoàn thành nếu thiếu đi sự hỗ trợ tận tình và bền bỉ từ thầy.

Tôi cũng xin gửi lời cảm ơn chân thành tới các đồng nghiệp của mình đã hỗ trợ tôi trong vấn đề cung cấp dữ liệu và hạ tầng training AI.

Cuối cùng, tôi xin dành lời tri ân sâu sắc đến gia đình – những người luôn bên cạnh, yêu thương và ủng hộ tôi vô điều kiện trong suốt chặng đường nghiên cứu này

.

Table of Contents

[**Acknowledgments 2**](#_heading=h.2et92p0)

[**Table of Contents 3**](#_heading=h.tyjcwt)

[**List of Figures 4**](#_heading=h.3dy6vkm)

[**Chapter 1 Introduction 5**](#_heading=h.1t3h5sf)

[**1**](#_heading=h.4d34og8) **Problem & Motivation 5**

[**2**](#_heading=h.17dp8vu) **Existing work 6**

[**3**](#_heading=h.3rdcrjn) **Problem & Motivation 7**

[**Chapter 2 Background Study 9**](#_heading=h.26in1rg)

[**1**](#_heading=h.lnxbz9) **Business of Fashion Brands 9**

[**2**](#_heading=h.1ksv4uv) **Modular Software Architecture 10**

[**3**](#_heading=h.44sinio) **ML Models in Production 11**

[**Chapter 3 System Design 13**](#_heading=h.2jxsxqh)

[**1**](#_heading=h.z337ya) **Business Flow 13**

[**2**](#_heading=h.1y810tw) **Design Diagrams 14**

[2.1](#_heading=h.4i7ojhp) Context Diagram 15

[2.2](#_heading=h.1ci93xb) Container Diagram 16

[2.3](#_heading=h.2bn6wsx) Component Diagram 18

[**Chapter 4 Implementation & Discussion 21**](#_heading=h.3as4poj)

[**1**](#_heading=h.1pxezwc) **Try-On & Catalog Module 21**

[**2**](#_heading=h.ihv636) **Try-on Integration into an E-commerce Application 26**

[**3**](#_heading=h.41mghml) **MLOps Implementation 29**

[**4**](#_heading=h.vx1227) **Microservice Implementation 31**

[**5**](#_heading=h.4f1mdlm) **Discussion, Future work & Conclusion 34**

[**References 38**](#_heading=h.2u6wntf)

List of Figures

[Figure 1. The design process in the Textile & Clothing industry. 10](#_heading=h.35nkun2)

[Figure 2. GARMENTO system main flow 14](#_heading=h.3j2qqm3)

[Figure 3. Context Diagram of the GARMENTO Virtual Try-on system. 15](#_heading=h.2xcytpi)

[Figure 4. The Container Diagram of the GARMENTO Virtual Try-on System. 17](#_heading=h.3whwml4)

[Figure 5. The Component Diagram of GARMENTO Virtual Try-on System. 19](#_heading=h.qsh70q)

[Figure 6. User Interface of Try-On function. 21](#_heading=h.49x2ik5)

[Figure 7. Google sign-on screen. 22](#_heading=h.2p2csry)

[Figure 8. Obtaining service-level credentials for the Try-On & Catalog module. 23](#_heading=h.147n2zr)

[Figure 9. Performing virtual try-on with preset. 24](#_heading=h.3o7alnk)

[Figure 10. Catalogue status transitions. 25](#_heading=h.23ckvvd)

[Figure 11. Virtual try-on plugin integrated into an example e-commerce web application. 27](#_heading=h.32hioqz)

[Figure 12. Sequence diagram for E-commerce Application integration. 28](#_heading=h.1hmsyys)

[Figure 13. Model Registry’s Sequence Diagram. 29](#_heading=h.2grqrue)

[Figure 14. Service decomposition in the GARMENTO system 31](#_heading=h.3fwokq0)

[Figure 15. High-level deployment diagram (non-UML) of the GARMENTO prototype. 33](#_heading=h.1v1yuxt)

1. Introduction

# Problem & Motivation

Trong môi trường doanh nghiệp hiện tại, việc theo dõi, nhu cầu phân bổ và đánh giá tiến độ công việc hiệu quả ngày càng lớn, đặc biệt với sự gia tăng về quy mô dự án và độ phức tạp trong quy trình vận hành. Mặc dù nhiều công cụ như Jira hay Trello đã được áp dụng rộng rãi, chúng vẫn chủ yếu mang tính chất lưu trữ và theo dõi thủ công, đòi hỏi sự can thiệp thường xuyên từ người dùng. Điều này khiến quá trình tổng hợp báo cáo, đề xuất công việc hoặc hướng dẫn nhân viên mới trở nên tốn thời gian và thiếu nhất quán.

Song song với đó, sự phát triển nhanh chóng của các mô hình ngôn ngữ lớn (LLMs) như ChatGPT, Grok, DeepSeek mở ra cơ hội mới trong việc tự động hóa các tác vụ quản lý công việc. DeepSeek là một dòng mô hình ngôn ngữ mã nguồn mở được huấn luyện trên tập dữ liệu quy mô hàng nghìn tỷ token, với năng lực suy luận được tăng cường thông qua các chiến lược huấn luyện nhiều giai đoạn (4). Tuy nhiên, để tận dụng hiệu quả các mô hình này trong môi trường doanh nghiệp cụ thể, việc tinh chỉnh (fine-tuning) dựa trên dữ liệu nội bộ là điều thiết yếu. Điều này đặt ra thách thức về hiệu quả tài nguyên, khả năng mở rộng và nguy cơ làm mất đi tri thức nền của mô hình gốc nếu không được huấn luyện đúng cách (2).

Từ nhu cầu thực tiễn, bài viết này này tập trung phát triển một hệ thống trí tuệ nhân tạo hỗ trợ doanh nghiệp trong việc theo dõi, phân tích và tối ưu hóa hoạt động vận hành nội bộ. Hệ thống dựa trên DeepSeek, một trong những mô hình LLM mã nguồn mở nổi bật hiện nay, được huấn luyện trên dữ liệu quy mô lớn, sở hữu khả năng suy luận ngữ cảnh tốt và có hỗ trợ tiếng việt. Tuy nhiên, để khai thác hiệu quả mô hình trong môi trường doanh nghiệp cụ thể, nơi có ngôn ngữ nội bộ, thuật ngữ chuyên ngành và đặc thù quy trình riêng biệt, việc fine-tuning là yếu tố bắt buộc. Fine-tune không chỉ giúp thích nghi với dữ liệu đặc thù của tổ chức mà còn mở ra khả năng tự động hóa hàng loạt quy trình như: tạo báo cáo tiến độ công việc, đề xuất nhiệm vụ phù hợp theo bối cảnh và định hướng nhân viên mới thông qua các hướng dẫn được cá nhân hóa.

Để giải quyết các bài toán trên mà vẫn đảm bảo hiệu quả tài nguyên và độ ổn định mô hình, bài viết đề xuất kỹ thuật LoRA (Low-Rank Adaptation) trong quá trình tinh chỉnh mô hình DeepSeek. LoRA là một phương pháp được đề xuất bởi (1). cho phép giảm thiểu đáng kể số lượng trọng số cần cập nhật bằng cách chèn thêm các ma trận hạng thấp vào trong cấu trúc mô hình đã huấn luyện sẵn. Nhờ đó, quá trình fine-tune có thể được thực hiện với chi phí bộ nhớ và tính toán thấp hơn hàng chục lần so với tinh chỉnh toàn bộ mô hình, nhưng vẫn giữ nguyên được hiệu năng ở mức cao. Vì các trọng số gốc được giữ nguyên, mô hình sau tinh chỉnh bằng LoRA có khả năng duy trì tốt các tri thức nền, qua đó hạn chế hiệu ứng catastrophic forgetting (2).

LoRA cũng mang lại tính linh hoạt cao cho doanh nghiệp: chỉ cần lưu trữ phần trọng số được chèn thêm (adapter) thay vì toàn bộ mô hình sau khi huấn luyện, giúp giảm chi phí lưu trữ và dễ dàng triển khai đa phiên bản cho các bộ phận khác nhau. Trong bài viết này, dữ liệu huấn luyện được lấy từ các nền tảng Jira, cùng với các tệp log nội bộ ghi nhận tiến trình làm việc, phản hồi nhân viên và lịch sử nhiệm vụ. Dữ liệu này được xử lý và cấu trúc lại theo định dạng chuẩn đầu vào của mô hình, từ đó phục vụ cho việc huấn luyện các adapter LoRA theo từng tác vụ chuyên biệt: tạo báo cáo tổng hợp công việc, đề xuất nhiệm vụ tương thích với vai trò, và hỗ trợ định hướng và hướng dẫn xử lý task cho nhân viên mới.

Việc tích hợp mô hình DeepSeek đã được tinh chỉnh vào hệ thống quản lý công việc sẽ không chỉ giúp tự động hóa nhiều khâu quan trọng, mà còn đóng vai trò như một “trợ lý nội bộ”, có khả năng hiểu ngữ cảnh, đề xuất hành động và hỗ trợ ra quyết định theo thời gian thực. Đây là bước tiến cần thiết nhằm nâng cao năng lực vận hành trong bối cảnh doanh nghiệp đang đối mặt với áp lực chuyển đổi số và tối ưu nguồn lực.

# Existing work

Trong vài năm gần đây, việc ứng dụng các mô hình ngôn ngữ lớn (LLM) vào hệ thống hỗ trợ doanh nghiệp đã nhận được nhiều sự quan tâm từ. Các mô hình như GPT, LLaMA, và gần đây là DeepSeek đã cho thấy tiềm năng trong việc xử lý ngôn ngữ tự nhiên một cách linh hoạt và chính xác, mở ra hướng phát triển cho các hệ thống tự động như trợ lý công việc, công cụ tổng hợp báo cáo, và gợi ý tác vụ thông minh. Tuy nhiên, để triển khai các mô hình này một cách hiệu quả trong môi trường doanh nghiệp cụ thể, yêu cầu về tinh chỉnh mô hình (fine-tuning) theo dữ liệu nội bộ trở nên cần thiết.

Một trong những công trình có ảnh hưởng lớn trong lĩnh vực tinh chỉnh LLM là nghiên cứu của (1), giới thiệu kỹ thuật **LoRA – Low-Rank Adaptation**. Phương pháp này giữ lại toàn bộ trọng số của mô hình gốc và chỉ huấn luyện thêm hai ma trận hạng thấp tại các lớp attention, giúp giảm mạnh chi phí bộ nhớ và tính toán trong quá trình fine-tune. LoRA được chứng minh có thể đạt chất lượng tương đương với fine-tuning toàn phần trong nhiều tác vụ NLP, đồng thời giảm nguy cơ làm mất đi các kiến thức nền đã học trước đó. Nhờ tính chất nhẹ và linh hoạt, LoRA đã trở thành nền tảng cho nhiều phương pháp mở rộng như QLoRA (5), áp dụng LoRA trên mô hình lượng tử hóa 4-bit để tinh chỉnh hiệu quả trên GPU bộ nhớ thấp.

Bên cạnh hiệu quả tính toán, một vấn đề khác trong fine-tuning là **hiện tượng catastrophic forgetting**,khi mô hình được tinh chỉnh nhiều lần hoặc cập nhật liên tục theo thời gian. (2) đã nghiên cứu mối quan hệ giữa số lượng bước huấn luyện và mức độ suy giảm kiến thức gốc, chỉ ra rằng ngay cả với các kỹ thuật như LoRA, nếu không có chiến lược quản lý thông tin hợp lý thì mô hình vẫn bị mất kiến thức nền nghiêm trọng. (6) cung cấp một khảo sát toàn diện về các phương pháp continual learning cho LLMs, bao gồm rehearsal, regularization, parameter isolation, và kết hợp các kỹ thuật như adapter hoặc LoRA để duy trì hiệu năng lâu dài.

Đối với những model LLM mã nguồn mở, DeepSeek là mô hình có tiềm năng cao, được thiết kế theo hướng hỗ trợ cộng đồng nghiên cứu triển khai mô hình hiệu quả trên dữ liệu tùy biến. Trong báo cáo kỹ thuật mới nhất, nhóm DeepSeek (4) đã huấn luyện các mô hình từ 1.3B đến 67B tham số với dữ liệu chất lượng cao, hỗ trợ đa ngôn ngữ và tối ưu cho khả năng suy luận logic. DeepSeek-R1, phiên bản fine-tuned reasoning model của họ, đạt kết quả tương đương với các mô hình thương mại như OpenAI GPT-3.5 trong nhiều tác vụ đòi hỏi lập luận nhiều bước (7). Đáng chú ý, DeepSeek được phát hành với giấy phép mã nguồn mở đầy đủ, tạo điều kiện thuận lợi cho việc áp dụng trong môi trường doanh nghiệp không có khả năng truy cập mô hình thương mại đắt đỏ.

Một số tài liệu thực nghiệm cũng đã chứng minh khả năng ứng dụng của DeepSeek vào các bài toán đặc thù. Ví dụ, trong tài liệu hướng dẫn của DataCamp (8) các tác giả đã thực hiện tinh chỉnh DeepSeek-R1 Distill (8B) bằng kỹ thuật LoRA để xây dựng chatbot y khoa có khả năng suy luận theo chuỗi (chain-of-thought). Mặc dù không liên quan trực tiếp đến tác vụ quản lý công việc, nhưng phương pháp tiền xử lý dữ liệu và chiến lược fine-tuning trong nghiên cứu này hoàn toàn có thể áp dụng cho các bài toán tương tự như tạo báo cáo, phân tích tiến độ, hoặc đề xuất nhiệm vụ trong doanh nghiệp.

Cuối cùng, để triển khai các mô hình tinh chỉnh này vào thực tế, một vấn đề khác là hạn chế tài nguyên phần cứng. (9) đã chứng minh khả năng tinh chỉnh mô hình LLM quy mô trung bình (1.3B–7B tham số) bằng kỹ thuật LoRA ngay trên CPU thông qua kỹ thuật tối ưu tính toán ma trận và lượng tử hóa nhẹ. Điều này đặc biệt quan trọng với các doanh nghiệp nhỏ hoặc tổ chức không có GPU chuyên dụng, giúp hạ thấp rào cản triển khai hệ thống AI trong nội bộ.

Các công trình hiện tại đã xây dựng nền móng quan trọng cho việc áp dụng LLM vào các hệ thống trợ lý doanh nghiệp. Tuy nhiên, vẫn còn thiếu những nghiên cứu cụ thể hóa hướng tiếp cận trong bài toán **quản lý công việc**, nơi đòi hỏi khả năng tích hợp dữ liệu nội bộ (như Jira, hệ thống chat), giữ ổn định tri thức nền và hoạt động hiệu quả trên nền tảng tài nguyên hạn chế. Bài viết này kế thừa các hướng tiếp cận đã được xác lập, đồng thời mở rộng bằng cách kết hợp DeepSeek, LoRA và dữ liệu công việc thực tế để xây dựng một hệ thống hỗ trợ quản lý phù hợp với nhu cầu của doanh nghiệp hiện đại.

# Contribution

Bài viết tập trung vào việc thiết kế và phát triển một hệ thống quản lý công việc dựa trên mô hình ngôn ngữ lớn DeepSeek, với các thành phần được tinh chỉnh bằng kỹ thuật Low-Rank Adaptation (LoRA) nhằm đảm bảo hiệu quả triển khai trong môi trường doanh nghiệp có tài nguyên hạn chế.

Bài viết đề xuất một kiến trúc hệ thống quản lý công việc AI phù hợp với bối cảnh doanh nghiệp vừa và nhỏ, trong đó mô hình ngôn ngữ được phân tách theo các tác vụ cụ thể như: sinh báo cáo tổng hợp, gợi ý hướng xử lý công việc, và tạo hướng dẫn nhân sự mới. Việc chia nhỏ mô hình theo chức năng, thay vì sử dụng một mô hình duy nhất, cho phép tối ưu hóa hiệu năng và khả năng bảo trì trong dài hạn.

Bài viết áp dụng kỹ thuật LoRA để tinh chỉnh mô hình DeepSeek-R1 với dữ liệu nội bộ từ Jira và dữ liệu chat nội bộ. Những nghiên cứu (1) (2) cho thấy LoRA không chỉ giảm đáng kể chi phí tính toán và lưu trữ mà còn giúp duy trì tri thức nền của mô hình, từ đó giảm thiểu hiện tượng "quên lãng thảm họa" thường gặp khi fine-tuning LLM. Việc áp dụng LoRA trong bài toán quản lý công việc là một điểm mới có giá trị thực tiễn.

Bài viết xây dựng một quy trình xử lý dữ liệu từ hệ thống doanh nghiệp sang định dạng huấn luyện cho LLM, bao gồm tiền xử lý, cấu trúc hóa log, ánh xạ tác vụ và gán nhãn cho từng tình huống. Đây là bước quan trọng nhằm chuyển đổi dữ liệu phi cấu trúc thành tri thức đầu vào cho mô hình ngôn ngữ, đồng thời mở ra khả năng tự động hóa các quy trình ra quyết định nội bộ.

Bài viết minh chứng cho việc thu hẹp khoảng cách giữa nghiên cứu học thuật và ứng dụng doanh nghiệp. Các kết quả và quy trình được xây dựng có thể tái sử dụng cho các hệ thống hỗ trợ khác, hoặc mở rộng sang các lĩnh vực như chăm sóc khách hang.

1. Background Study

Befo

# Business Workflow in Enterprises

Công cụ thường dùng như Jira, Trello, GitLab

Các vấn đề phổ biến:

Nhu cầu tự động hóa

X2

# Modular Software Architecture

Microservices

X1

# ML Models in Production

training → fine-tuning → deployment → monitoring

Problem: lightweight deployment, huấn luyện lại theo feedback, Risk???

X2-x3

1. System Design

Befo

# Business Flow

First of

# Design Diagrams

A so

## Context Diagram

The

## Container Diagram

The

## Component Diagram

A

1. Implementation & Discussion

This .

# Try-On & Catalog Module

In

# Try-on Integration into an E-commerce Application

Integration

# MLOps Implementation

There a

# Discussion, Future work & Conclusion

This

Appendix

**Appendix A**. System test cases

List of test cases used in the development of the GARMENTO system is available publicly at: <https://bit.ly/3W4mXUM>. Currently, all critical paths have been tested and passed all the test cases.

**Appendix B.** Latest source code directory

In addition to the monorepo provided in the References list, we provide an organization that clearly shows the decomposed and developed microservices. The Github organization is available at: <https://github.com/garmento-microservices>.

References