**TRƯỜNG ĐẠI HỌC QUỐC TẾ HỒNG BÀNG**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

**🙢🕮🙠**



**BÁO CÁO ĐỒ ÁN THỰC TẬP**

**MÔN HỌC  
THỰC TẬP DOANH NGHIỆP**

Đề tài: Summarization Studio – Xưởng tóm tắt thông minh theo khối lớp

**Giảng viên hướng dẫn: Hoàng Ngọc Long**

**Sinh viên thực hiện: Nguyễn Minh Vũ**

**Mã số sinh viên: 2211110063**

**TP. Hồ Chí Minh, 2025**

LỜI CẢM ƠN

Để hoàn thành đề tài này, em đã nhận được sự hướng dẫn, giúp đỡ và góp ý nhiệt tình của quý thầy cô trường Đại Học Quốc Tế Hồng Bàng và thầy Hoàng Ngọc Long.

Em xin gửi lời biết ơn sâu sắc đến thầy Hoàng Ngọc Long đã dành nhiều thời gian và tâm huyết hướng dẫn nghiên cứu và giúp em hoàn thành môn học.

Em cũng xin chân thành cảm ơn đến quý thầy cô trường Đại học Quốc Tế Hồng Bàng, đặc biệt là những thầy cô đã tận tình dạy bảo cho em suốt thời gian học tập tại trường.

Em đã có nhiều cố gắng hoàn thiện dự án bằng tất cả năng lực của mình, tuy nhiên không thể tránh khỏi nhiều thiếu sót, rất mong nhận được những đóng góp quý báu của quý thầy cô và các bạn.

TP.HCM, ... Tháng ... Năm 2025

Người thực hiện

Nguyễn Minh Vũ

TRANG CAM KẾT

Tôi xin cam kết báo cáo thường kỳ này được hoàn thành dựa trên các kết quả thực hiện bài thực hành của tôi và các mã nguồn và kết quả này chưa được dùng cho bất cứ báo cáo của sinh viên nào khác.

*TP.HCM, ngày ... tháng ... năm 2025*

Người thực hiện

Nguyễn Minh Vũ

NHẬN XÉT CỦA GIẢNG VIÊN

TP.HCM, Ngày ... Tháng ... Năm 2025

Chữ ký giảng viên

DANH MỤC BIỂU ĐỒ HÌNH VẼ

DANH MỤC BIỂU ĐỒ BẢNG

DANH MỤC THUẬT NGỮ VIẾT TẮT

MỤC LỤC

MỞ ĐẦU

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

Chương này giới thiệu bài toán tóm tắt văn bản tự động và nhu cầu hỗ trợ học sinh tiểu học đọc hiểu tốt hơn. Nghiên cứu xác định khoảng trống trong việc áp dụng công nghệ xử lý ngôn ngữ tự nhiên vào giáo dục, đề xuất xây dựng hệ thống tóm tắt bằng hai phương pháp: trích xuất và diễn giải. Mục tiêu là giúp học sinh dễ dàng nắm nội dung chính, đồng thời làm rõ ý nghĩa khoa học và thực tiễn của đề tài qua phân tích các nghiên cứu liên quan.

## Tổng quan

Tóm tắt văn bản tự động là một bài toán cốt lõi trong xử lý ngôn ngữ tự nhiên (NLP), giúp rút gọn thông tin để hỗ trợ học tập và nghiên cứu. Quá trình này bắt nguồn từ ý tưởng trích xuất ý chính từ văn bản dựa trên từ khóa, đặt nền móng cho các phương pháp tóm tắt hiện đại. Tóm tắt được thực hiện theo hai hướng chính: tóm tắt trích xuất và tóm tắt diễn giải.

* **Tóm tắt trích xuất (*Extractive Summarization***): Sử dụng thuật toán thống kê và học máy để chọn các câu quan trọng nhất từ văn bản gốc, dựa trên điểm số từ các tiêu chí như tần suất từ khóa, vị trí câu và ngữ nghĩa. Các câu có điểm cao được giữ nguyên để tạo bản tóm tắt.
* **Tóm tắt diễn giải (*Abstractive Summarization*):** Mô phỏng cách con người tóm tắt, phương pháp này dùng mô hình ngôn ngữ để sinh ra câu mới, diễn đạt ý chính ngắn gọn và tự nhiên.

Trong dự án này, hệ thống đa tác tử (MAS) được thiết kế dựa trên hướng phát triển của hai cách tóm tắt trích xuất và diễn giải. Nhằm biến bản thiết kế lý thuyết thành một hệ thống hoạt động thực tế, có khả năng tự động nhận diện, tóm tắt và đánh giá chất lượng bài đọc tiếng Việt cho học sinh tiểu học. Hệ thống hoàn chỉnh giúp tự động hóa quy trình tóm tắt bài đọc theo nhiều cấp độ, đảm bảo tính minh bạch, kiểm soát được độ khó và từ vựng theo khối lớp, đồng thời cung cấp công cụ hữu ích cho giáo viên trong việc kiểm tra và điều chỉnh nội dung học tập.

Mục tiêu là xây dựng hệ thống tóm tắt văn bản dành cho học sinh tiểu học, giúp các em dễ dàng hiểu truyện và bài đọc, từ đó nâng cao hiệu quả học tập và ứng dụng tóm tắt tiếng Việt trong giáo dục.

## Lý do chọn đề tài

Trong bối cảnh giáo dục hiện đại, nhu cầu về các công cụ hỗ trợ học tập cho học sinh tiểu học ngày càng trở nên cần thiết. Khối lượng thông tin lớn từ sách vở và tài liệu khiến trẻ phải tiếp cận nhiều nội dung, nhưng khả năng tập trung và đọc hiểu của các em còn hạn chế. Học sinh tiểu học thường gặp khó khăn khi xử lý các văn bản dài, dẫn đến hiệu quả học tập giảm sút.

Trước thực tế đó, việc xây dựng một hệ thống hỗ trợ tóm tắt truyện và bài đọc tự động, thân thiện và phù hợp với lứa tuổi tiểu học là cần thiết. Hệ thống này không chỉ giúp học sinh rèn luyện kỹ năng đọc hiểu mà còn hỗ trợ giáo viên và phụ huynh trong việc kiểm tra, đánh giá khả năng tiếp thu của học sinh.

Bên cạnh đó, sự phát triển của các mô hình ngôn ngữ tự nhiên (NLP) và trí tuệ nhân tạo (AI) hiện nay như PhoBERT, mT5, GPT, BART, hoặc các mô hình Transformer khác mở ra cơ hội lớn cho việc ứng dụng công nghệ vào giáo dục. Việc triển khai một hệ thống tóm tắt dựa trên AI không chỉ mang ý nghĩa học thuật mà còn có giá trị thực tiễn cao, góp phần đổi mới phương pháp học tập và giảng dạy theo hướng thông minh, hiện đại và cá nhân hóa.

Xuất phát từ thực tế này, dự án ứng dụng các mô hình LLMs và AI vào việc nghiên cứu xây dựng hệ thống tóm tắt giúp trẻ vượt qua rào cản đọc hiểu, dự án là một công cụ hữu ích, góp phần nâng cao hiệu quả học tập và khơi dậy hứng thú đọc hiểu cho học sinh. Đồng thời mở rộng ứng dụng NLP và AI vào giáo dục, tối ưu hóa quá trình tiếp thu thông tin cho học sinh.

## Mục tiêu nghiên cứu

Xây dựng một hệ thống tóm tắt tự động có khả năng rút gọn nội dung truyện và bài đọc tiếng Việt sao cho ngắn gọn, dễ hiểu, phù hợp với lứa tuổi học sinh tiểu học, hỗ trợ quá trình học tập, đọc hiểu và ôn luyện kiến thức.

* Phân tích yêu cầu và đặc thù nội dung của các truyện, bài đọc dành cho học sinh tiểu học (về độ dài, ngôn ngữ, cấu trúc, từ vựng, độ khó).
* Thiết kế và xây dựng hệ thống tóm tắt kết hợp hai phương pháp chính:
* Tóm tắt trích xuất (extractive summarization) – lựa chọn các câu quan trọng trong văn bản gốc.
* Tóm tắt diễn giải (abstractive summarization) – tạo ra câu mới thể hiện lại ý chính bằng ngôn ngữ tự nhiên.
* Fine-tune và đánh giá các mô hình ngôn ngữ (ví dụ: PhoBERT, mT5, hoặc mô hình LLM khác) để tối ưu kết quả tóm tắt cho tiếng Việt.
* Triển khai hệ thống ứng dụng (web hoặc mobile app) cho phép người dùng nhập nội dung truyện/bài đọc và nhận kết quả tóm tắt tức thời.
* Đánh giá chất lượng tóm tắt theo các tiêu chí như: độ chính xác ngữ nghĩa, mức độ ngắn gọn, tính tự nhiên, và mức độ phù hợp với lứa tuổi học sinh.

## Ý nghĩa đề tài

### Ý nghĩa khoa học

* Góp phần nghiên cứu và ứng dụng các mô hình ngôn ngữ hiện đại (LLMs) vào lĩnh vực xử lý tiếng Việt, đặc biệt là nhiệm vụ tóm tắt văn bản — một bài toán phức tạp trong xử lý ngôn ngữ tự nhiên (NLP).
* Cung cấp một mô hình, hệ thống kết hợp giữa tóm tắt trích xuất và tóm tắt diễn giải, giúp cải thiện độ chính xác và tính tự nhiên của bản tóm tắt.
* Mở rộng nguồn dữ liệu và phương pháp huấn luyện mô hình ngôn ngữ tiếng Việt, đóng góp cho cộng đồng nghiên cứu AI và NLP trong nước.
* Làm cơ sở cho các nghiên cứu tiếp theo về tóm tắt văn bản, sinh câu hỏi tự động, hoặc rút trích bài học từ nội dung giáo dục.

### Ý nghĩa thực tiễn

* Giúp học sinh tiểu học dễ dàng nắm bắt nội dung chính của truyện và bài đọc, phát triển kỹ năng đọc hiểu và tư duy tổng hợp.
* Hỗ trợ giáo viên và phụ huynh trong việc kiểm tra, giảng dạy và định hướng học tập cho học sinh thông qua các bản tóm tắt ngắn gọn, phù hợp với độ tuổi.
* Tiết kiệm thời gian cho người học trong việc nắm bắt nhanh kiến thức, đồng thời tăng hứng thú đọc sách và học tiếng Việt.
* Có thể mở rộng ứng dụng trong giáo dục thông minh, sách điện tử, hoặc nền tảng học tập trực tuyến, góp phần thúc đẩy chuyển đổi số trong giáo dục Việt Nam.

## Định hướng giải pháp

### Định hướng chung

Hệ thống được xây dựng theo mô hình đa tác tử (Multi-Agent System - MAS), trong đó mỗi agent đảm nhận một vai trò cụ thể trong quá trình phân tích – hiểu – tóm tắt – đánh giá văn bản.

Cách tiếp cận này giúp hệ thống phân rã nhiệm vụ phức tạp thành nhiều bước nhỏ, hoạt động độc lập nhưng có phối hợp, tương tự như cách con người đọc, hiểu, và rút ý chính từ nội dung.

### Định hướng triển khai

* Ngôn ngữ lập trình: Python.
* Framework đề xuất: LangGraph hoặc LangChain để xây dựng pipeline giữa các agent.
* Mô hình ngôn ngữ sử dụng: PhoBERT (trích xuất), mT5 (diễn giải), Ollama – llama3:8b (Sử dụng cho việc demo flow System).
* Giao diện: xây dựng bằng React kết hợp với Spring Boot kết hợp Flask.
* Lưu trữ dữ liệu: MySQL cho hệ thống backend và Chroma cho phiên lưu dữ liệu của MAS.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



## Giới thiệu về tóm tắt văn bản

### Khái niệm

Tóm tắt văn bản là quá trình rút gọn nội dung của một văn bản gốc thành một phiên bản ngắn hơn nhưng vẫn giữ lại được các ý chính, nội dung cốt lõi và thông điệp quan trọng. Mục tiêu của tóm tắt là giúp người đọc hiểu nhanh nội dung chính mà không cần đọc toàn bộ văn bản ban đầu.

Trong lĩnh vực xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP), tóm tắt văn bản là một trong những bài toán quan trọng và thách thức, bởi nó đòi hỏi hệ thống phải có khả năng hiểu ngữ nghĩa, ngữ pháp và bối cảnh của văn bản.

### Phân loại

* Tóm tắt trích xuất

Phương pháp này chọn trực tiếp các câu quan trọng trong văn bản gốc để tạo thành bản tóm tắt. Không sinh câu mới mà chỉ chọn lọc và sắp xếp lại những câu có giá trị thông tin cao.

Các kỹ thuật phổ biến:

* Thống kê từ khóa: TF-IDF, độ quan trọng của câu.
* Mô hình đồ thị: TextRank, LexRank.
* Mô hình ngôn ngữ: BERT, PhoBERT.

Ưu điểm: đơn giản, nhanh, dễ triển khai.

Nhược điểm: bản tóm tắt có thể thiếu tính tự nhiên và chưa mạch lạc.

* Tóm tắt diễn giải (Abstractive Summarization)

Phương pháp này hiểu nội dung văn bản gốc rồi diễn đạt lại bằng ngôn ngữ tự nhiên mới, giống như con người viết lại bằng lời của mình. Thường sử dụng các mô hình học sâu (Deep Learning) như Transformer, BART, T5, mT5, GPT.

Ưu điểm: kết quả tự nhiên, mạch lạc, có thể rút gọn mạnh hơn.

Nhược điểm: phức tạp, đòi hỏi dữ liệu huấn luyện lớn và mô hình mạnh.

### Các ứng dụng

1. Giáo dục: tạo tóm tắt bài học, giúp học sinh nắm nội dung nhanh hơn.
2. Báo chí: sinh tóm tắt tự động cho tin tức, báo mạng.
3. Nghiên cứu: tóm tắt tài liệu học thuật, báo cáo khoa học.
4. Trợ lý ảo: tổng hợp nội dung email, hội thoại hoặc văn bản dài.
5. Y tế – pháp luật – hành chính: rút gọn văn bản, biên bản, hoặc hồ sơ.

### Vai trò của tóm tắt trong giáo dục tiểu học

Trong bối cảnh giáo dục hiện nay, việc áp dụng tóm tắt tự động giúp:

* Học sinh hiểu nhanh nội dung truyện, bài đọc mà không bị quá tải thông tin.
* Giáo viên tạo tài liệu học tập, bài giảng rút gọn hiệu quả hơn.
* Thúc đẩy việc ứng dụng trí tuệ nhân tạo vào giáo dục, hỗ trợ quá trình học tập chủ động và thông minh.

## Tổng quan về xử lý ngôn ngữ tự nhiên

### Khái niệm

Xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP) là một lĩnh vực thuộc trí tuệ nhân tạo (Artificial Intelligence – AI) và khoa học máy tính, nghiên cứu các phương pháp giúp máy tính có thể hiểu, phân tích, và sinh ngôn ngữ của con người một cách tự nhiên.

Mục tiêu của NLP là tạo ra các hệ thống có khả năng:

* Hiểu được nội dung và ngữ nghĩa trong NLP (tiếng nói, văn bản).
* Tương tác với con người thông qua ngôn ngữ nói hoặc viết.
* Sinh ngôn ngữ tự nhiên để trả lời, tóm tắt, dịch, hoặc mô tả thông tin.

Nói cách khác, NLP là cầu nối giữa ngôn ngữ con người và máy tính, giúp máy có thể xử lý dữ liệu phi cấu trúc (như văn bản, hội thoại, tin tức, tài liệu).

### NLP cho tiếng Việt

Việc xử lý tiếng Việt gặp nhiều thách thức do:

* Đặc điểm ngôn ngữ phức tạp: nhiều từ ghép, dấu thanh, và hiện tượng đồng âm dị nghĩa.
* Tách từ khó: không có dấu cách rõ ràng giữa các từ.
* Thiếu dữ liệu lớn và gán nhãn sẵn so với tiếng Anh.

Một số công cụ và mô hình NLP tiếng Việt phổ biến:

* Thư viện tiền xử lý: Underthesea, PyVi, VnCoreNLP.
* Mô hình ngôn ngữ: PhoBERT, ViT5, mT5, BARTpho.
* Ứng dụng: tóm tắt văn bản, nhận dạng thực thể, phân tích cảm xúc, chatbot tiếng Việt.

### Vai trò của NLP trong hệ thống tóm tắt truyện và bài đọc

Trong đề tài này, NLP đóng vai trò nền tảng để:

* Hiểu và phân tích ngữ nghĩa câu chuyện.
* Xác định ý chính, nhân vật, và mối quan hệ giữa các đoạn.
* Tạo ra bản tóm tắt ngắn gọn, mạch lạc, phù hợp lứa tuổi học sinh.
* Hỗ trợ các tác tử trong hệ thống Multi-Agent (như OCR Reader, Extractor, Abstracter, Evaluator, …) giao tiếp và xử lý ngôn ngữ một cách thống nhất.

## Mô hình LLMs và kiến trúc Transformer

### Tổng quan về LLMs (Large Language Models)

Large Language Models (LLMs) là các mô hình học sâu được huấn luyện trên lượng dữ liệu văn bản khổng lồ nhằm học và hiểu ngữ nghĩa, cú pháp, cũng như các mối quan hệ ngôn ngữ tự nhiên. Nguyên lý hoạt động dựa trên kiến trúc transformer.

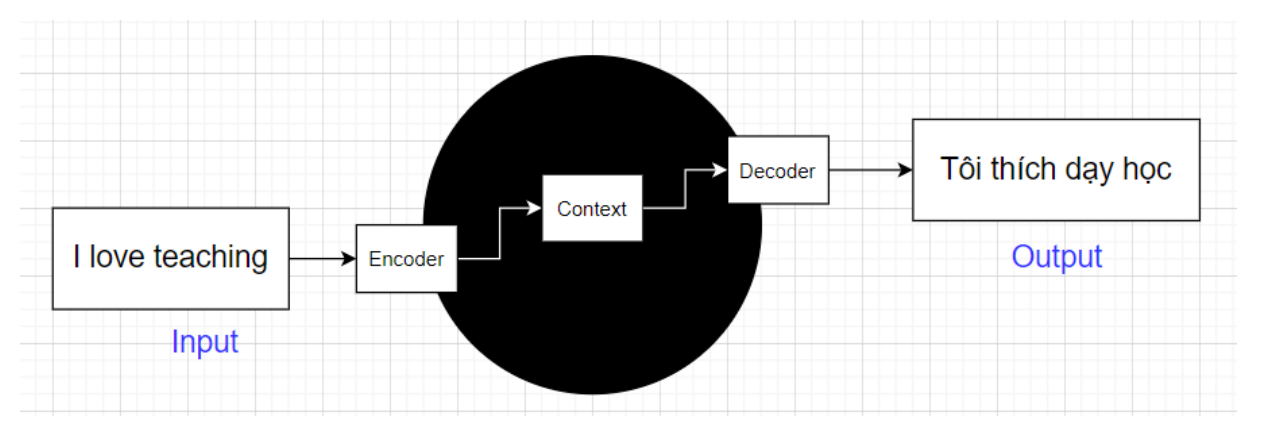
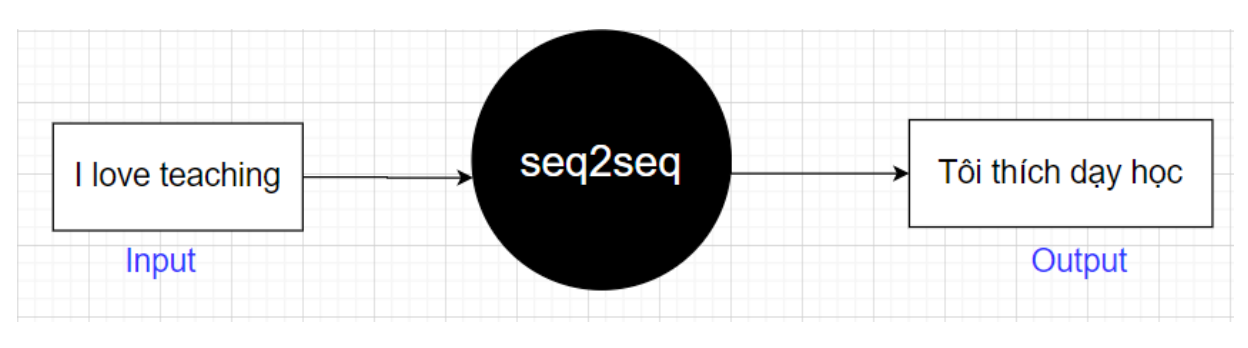
Các mô hình này có khả năng:

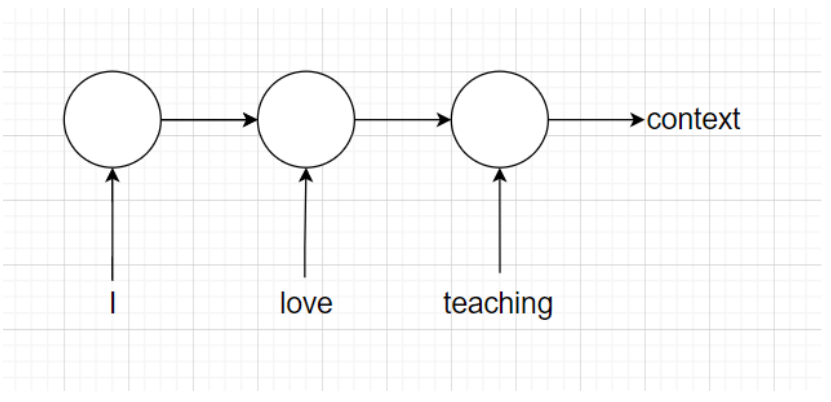
* Sinh văn bản tự nhiên (text generation).
* Hiểu và tóm tắt văn bản.
* Trả lời câu hỏi, dịch ngôn ngữ, hoặc chuyển đổi truy vấn tự nhiên sang ngôn ngữ lập trình (như SQL).

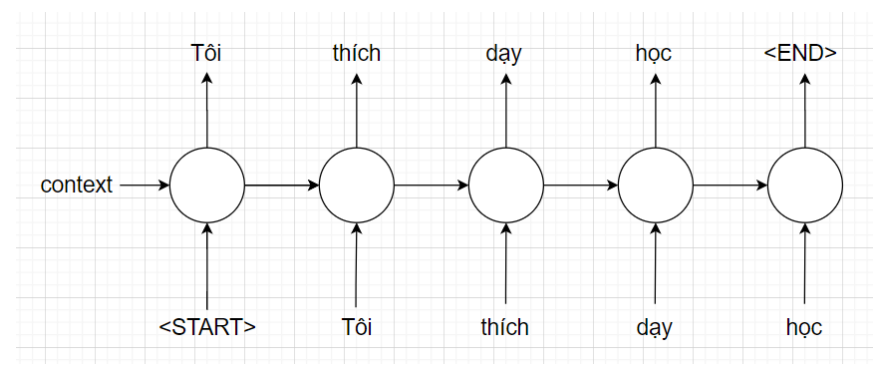
Các LLM nổi bật hiện nay gồm:

* GPT (Generative Pre-trained Transformer) của OpenAI.
* T5 / mT5 (Text-to-Text Transfer Transformer) của Google.
* LLaMA của Meta.
* Phi-3, Qwen2, Mistral, Claude, Gemini, v.v.

Ví dụ điển hình là mô hình seq2seq:









### Kiến trúc Transformer

Transformer là kiến trúc mạng nơ-ron, chúng nền tảng của hầu hết các LLM hiện nay, được giới thiệu lần đầu bởi Vaswani et al. (2017) trong bài báo “Attention Is All You Need”. Kiến trúc này thay thế hoàn toàn cơ chế RNN và LSTM truyền thống bằng cơ chế Self-Attention.

* Thành phần chính của Transformer:
* Input Embedding: Biến đổi các từ thành vector số (word embeddings).
* Positional Encoding: Thêm thông tin vị trí của từ trong câu để mô hình hiểu thứ tự.
* Encoder: Gồm nhiều lớp self-attention và feed-forward network, mã hóa ngữ cảnh của đầu vào.
* Decoder: Dùng cơ chế attention để sinh đầu ra dựa trên thông tin từ encoder.
* Multi-Head Attention: Cho phép mô hình học nhiều mối quan hệ giữa các từ trong cùng một lúc.

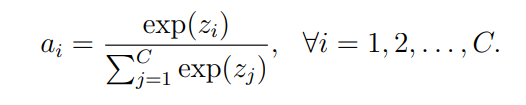
Công thức tính Attention:

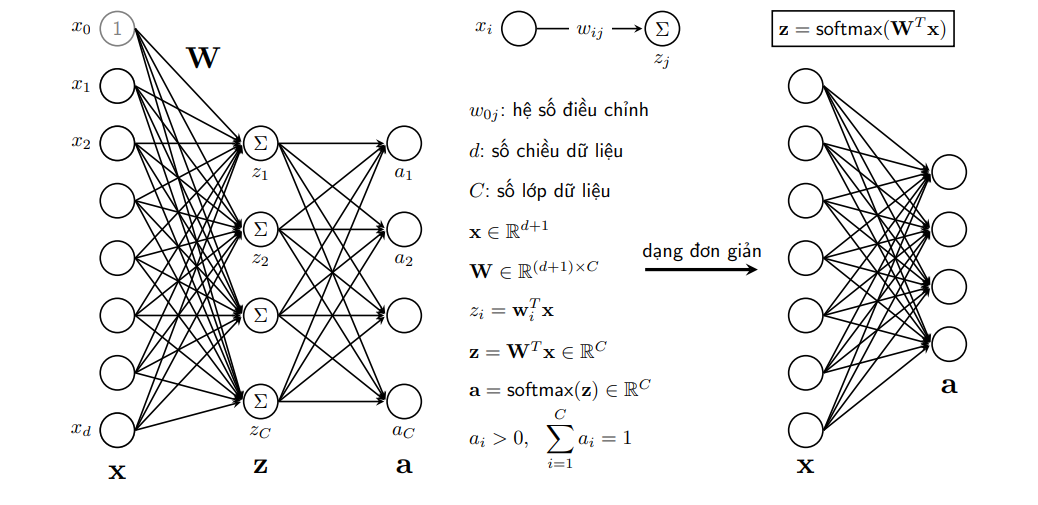
Attention(Q, K, V) = Softmax

Trong đó:

* Q,K,V lần lượt là ma trận Query, Key, và Value.
* là kích thước vector Key.

Công thức tính Softmax:





### LLMs dựa trên Transformer

Các mô hình LLM hiện đại thường mở rộng Transformer bằng:

* Huấn luyện trên quy mô dữ liệu cực lớn (hàng trăm tỷ token).
* Sử dụng cơ chế attention tối ưu hóa (như FlashAttention, Rotary Embedding).
* Tối ưu bộ nhớ và tốc độ suy luận thông qua kỹ thuật như quantization hoặc LoRA fine-tuning.

Ví dụ:

* mT5 (Google): mô hình đa ngôn ngữ dựa trên Transformer Encoder–Decoder.
* PhoBERT (VinAI): mô hình Transformer Encoder tối ưu cho tiếng Việt.
* Qwen2.5, GPT-4, Gemini 2.5: các mô hình LLM thế hệ mới dựa hoàn toàn trên cải tiến của Transformer.

### Ưu điểm của kiến trúc Transformer trong bài toán tóm tắt

1. Có khả năng hiểu ngữ cảnh toàn cục thay vì tuần tự như RNN.
2. Dễ dàng fine-tune cho các nhiệm vụ cụ thể (tóm tắt, dịch, sinh SQL, v.v.).
3. Cho phép xử lý song song trên GPU, giúp rút ngắn thời gian huấn luyện.
4. Phù hợp với ngôn ngữ tự nhiên tiếng Việt, đặc biệt khi kết hợp mô hình như mT5 hoặc PhoBERT.

## Lý thuyết về đa tác tử (Multi-Agent System)

### Single AI Agent

#### Khái niệm

Trong ngữ cảnh máy tính hay tự động hóa, “Agent” (tác tử) là một hệ thống hoặc chương trình máy tính có khả năng nhận thức môi trường thông qua cảm biến, xử lý thông tin và thực hiện hành động để đạt được mục tiêu cụ thể. Các Agent có thể hoạt động độc lập hoặc tương tác với người dùng và môi trường xung quanh.

#### Đặc điểm của AI Agent

* Nhận thức môi trường: Sử dụng cảm biến để thu thập thông tin từ môi trường.
* Ra quyết định: Phân tích thông tin và đưa ra hành động phù hợp.
* Xây dựng các quy tắc hoặc thuật toán xác định hành động.

Ví dụ: Nếu nhiệt độ vượt quá 30 độ 🡪 bật quạt; nếu mức nước giảm dưới 20% 🡪 thông báo cho người sử dụng.

* Thực hiện hành động: Áp dụng các hành động đã quyết định để đạt được mục tiêu.
* Gọi thực hiện hành động theo cơ chế API, hoặc chạy chương trình máy tính,…
* Tự động hóa: Có thể hoạt động mà không cần sự can thiệp trực tiếp của con người.

#### Hạn chế của AI Agent

* Hoạt động theo bộ quy tắc đã định sẵn.
* Chỉ áp dụng các trường hợp đơn giản (ít tham số / môi trường đơn giản), các quy tắc định sẵn.
* Có mục tiêu cụ thể và chức năng hạn chế.
* Một số tình huống không thể dùng quy tắc (nhiều tham số / môi trường phức tạp).
* Khả năng thích nghi và học hỏi rất thấp, không xử lý tốt tình huống ngoài dự đoán.

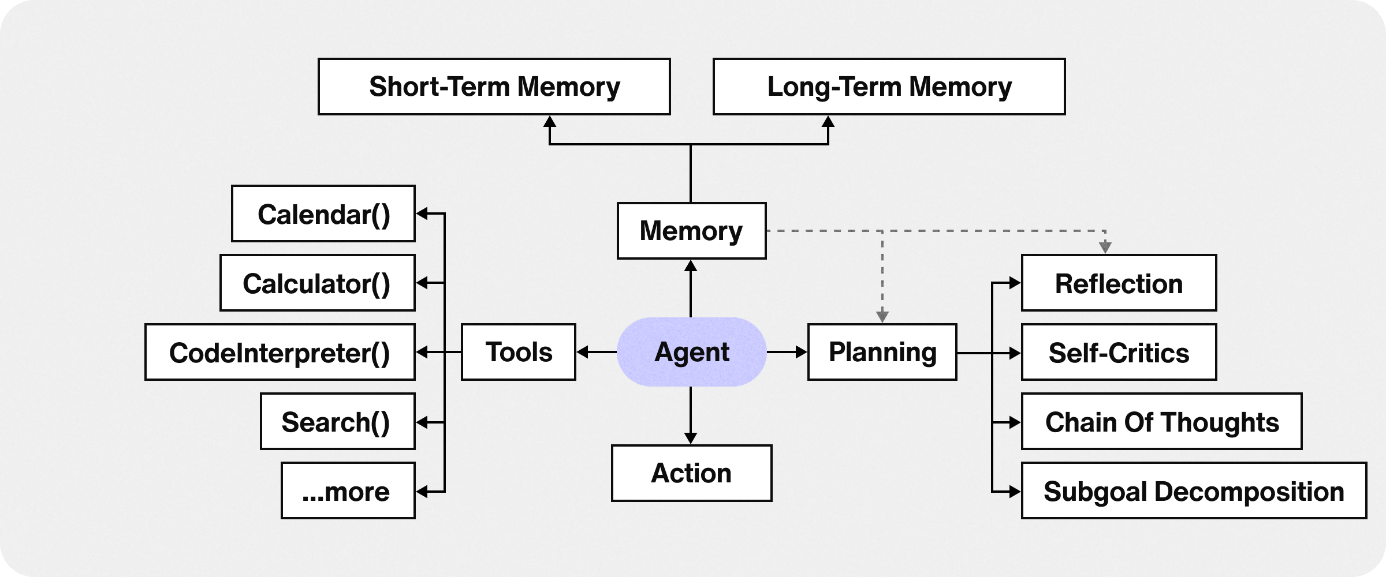
#### Lợi ích và hiệu quả mang lại của hệ thống

* Khả năng suy nghĩ, phân tích.
* Đưa ra kế hoạch thực thi.
* Đưa ra quyết định.
* Thực hiện hành động.
* Phản ánh khi có phản hồi từ môi trường.
* Đưa ra hành động tiếp theo.
* Học từ những phản hồi với môi trường.

Tác tử AI là một chường trình nhận đầu vào, suy nghĩ và hành động để hoàn thành một nhiệm vụ bằng cách sử dụng công cụ, trí nhớ và tri thức. Có suy nghĩ và hành động như con người.

#### Kiến trúc của AI Agent

* Phản ánh → nhìn lại để học hỏi.
* Tự phê bình → tự đánh giá, tránh sai lầm.
* Chuỗi suy nghĩ → lập luận theo bước, dễ kiểm soát.
* Phân tích mục tiêu phụ → chia nhỏ vấn đề để nhiều agent phối hợp hiệu quả.



#### Ví dụ về quá trình hoạt động của AI Agent

|  |  |  |
| --- | --- | --- |
| User | Tìm vé máy bay? | |
| AI Agent | | * Thought: Câu hỏi này hỏi này chưa đủ thông tin, yêu cầu người dùng bổ sung. * Response: Bạn muốn đi từ đâu, đến đâu, vào ngày nào ? |
| User | | Từ Hà Nội |
| AI Agent | | * Thought: Chưa đủ thông tin, yêu cầu người dùng bổ sung. * Response: Bạn muốn đến đâu, vào ngày nào ? |
| User | | Đến TP. Hồ Chí Minh, ngày 7/8/2025 |
| AI Agent | | * Thought: Đã đủ thông tin, cần công cụ tìm chuyến bay. * Action: <<function\_call>>  search\_flight(...) * Observation: API Result: [{“time”: “...”, “price”: “...”}] * Thought: Danh sách chuyến bay là ... * Response: “Danh sách chuyến bay là à trời nắng”. |

**ReAct Prompting**: Giúp cho LLMs tích hợp được các phân tích, đưa ra những hành động cần thực hiện theo một cách có cấu trúc.

#### Giải pháp xây dựng

AI Agent: Kết hợp điểm mạnh của LLMs và các công cụ.

* Khả năng phân tích xử lý ngôn ngữ tự nhiên mạnh mẽ
* Có khả năng phân tích và hiểu yêu cầu người dùng.
* Thực hiện được các thao tác phân tích văn bản.
* Ghi nhớ và tổng hợp kiến thức từ dữ liệu huấn luyện.
* Đưa ra kế hoạch thực thi.
* Tổng hợp phân tích kết quả.
* Khả năng truy xuất của những công cụ với nguồn dữ liệu bên ngoài
* API truy xuất dữ liệu bên ngoài
* Các công cụ tìm kiếm
* Công cụ thực thi mã nguồn
* API điều khiển thiết bị

### Multi-Agent

#### Khái niệm

Hệ thống đa tác tử (Multi-Agent System – MAS) là một mô hình tính toán bao gồm nhiều tác tử (agent) cùng hoạt động trong một môi trường chung, mỗi tác tử có khả năng quan sát, suy luận, ra quyết định và tương tác với các tác tử khác để hoàn thành mục tiêu riêng hoặc chung của toàn hệ thống.

Mỗi tác tử trong MAS thường có các đặc tính cơ bản sau:

* Tự chủ (Autonomy): Tác tử có khả năng hoạt động độc lập mà không cần điều khiển trực tiếp.
* Nhận thức (Perception): Có khả năng cảm nhận trạng thái của môi trường xung quanh.
* Hành động (Action): Có thể thực hiện hành vi nhằm thay đổi trạng thái môi trường hoặc đạt mục tiêu cụ thể.
* Tương tác (Interaction): Có thể trao đổi thông tin, cộng tác hoặc cạnh tranh với các tác tử khác.

#### Cấu trúc của hệ thống MAS

Một hệ thống đa tác tử thường bao gồm ba thành phần chính:

* + 1. Tác tử (Agents): Các thực thể thông minh có mục tiêu và khả năng xử lý độc lập.
    2. Môi trường (Environment): Không gian mà trong đó các tác tử hoạt động và tác động lẫn nhau.
    3. Cơ chế giao tiếp (Communication Mechanism): Cho phép các tác tử chia sẻ thông tin hoặc phối hợp hành động (thường sử dụng giao thức như ACL – Agent Communication Language).

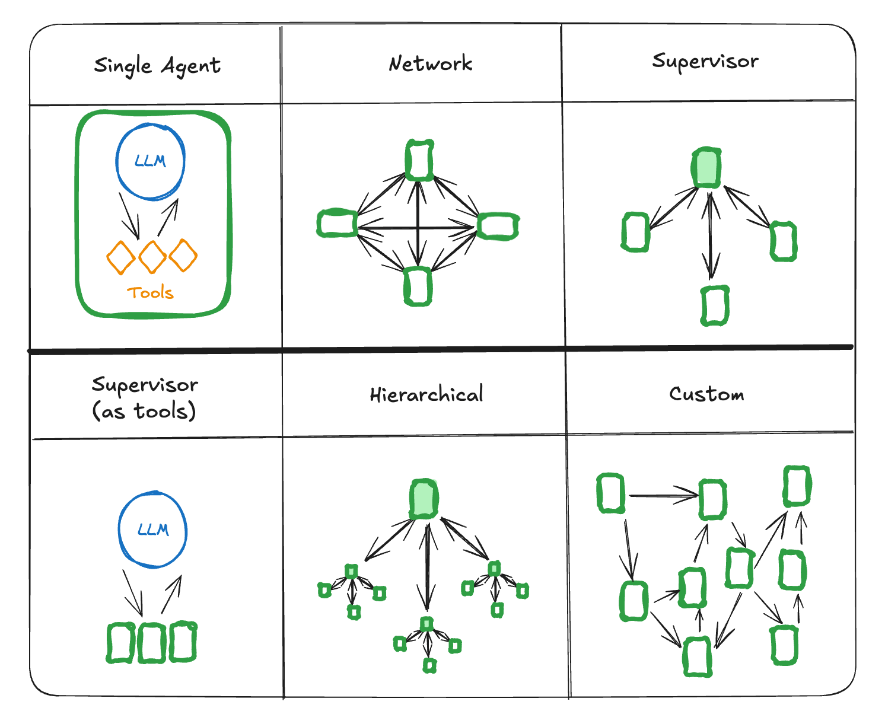
Mối quan hệ giữa các tác tử có thể mang tính:

1. Hợp tác (Cooperative): Các tác tử phối hợp để đạt được mục tiêu chung.
2. Cạnh tranh (Competitive): Các tác tử có mục tiêu xung đột, cố gắng tối ưu lợi ích riêng.
3. Lai (Hybrid): Kết hợp cả hai dạng trên tùy vào ngữ cảnh nhiệm vụ.

##### Giao tiếp trong hệ thống MAS

Các cách giao tiếp giữa agent với agent:

* Network: Mỗi tác tử có thể giao tiếp trực tiếp, tự do lựa chọn tác tử kế tiếp.
* Supervisor: Mỗi tác tử giao tiếp với 1 tác tử quản lý, agent quản lý sẽ quyết định sẽ gọi agent kế tiếp.
* Supervisor (tools): Các tác tử có thể được xem là các tools, supervisor quyết định gọi các tác tử và truyền tham số tương ứng.
* Hierarchical: Là trường hợp tổng quát của kiến trúc supervisor.
* Custom: Tùy biến, mỗi tác tử có thể kết nối đến một vài tác tử, tác tử sẽ tự quyết định tác tử nào sẽ được gọi kế tiếp.



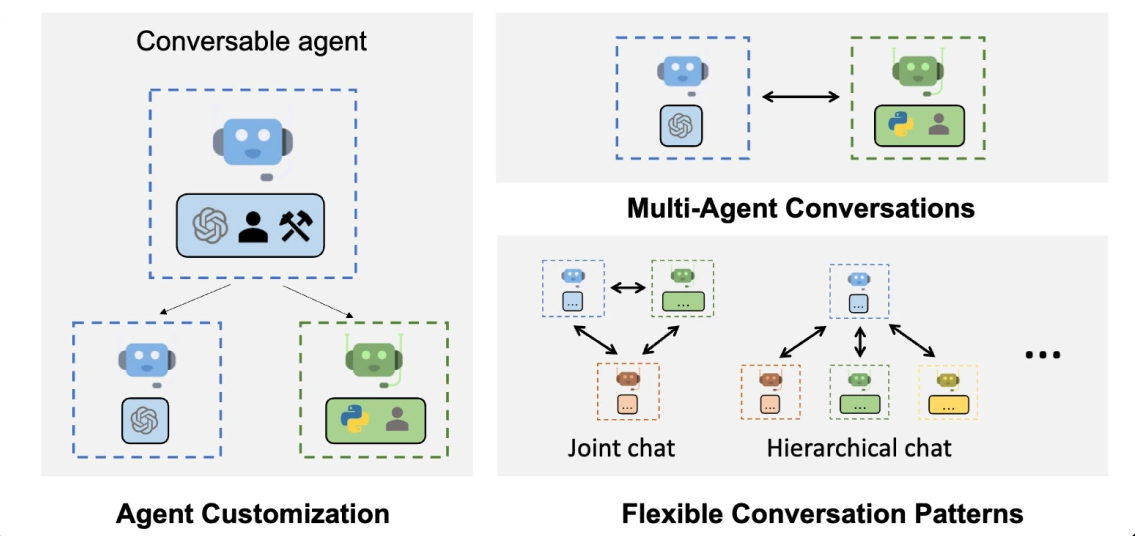
Trong các hệ thống đa tác tử, handoffs và tool calls là hai cách chính để các tác tử giao tiếp và chuyển giao quyền kiểm soát cho nhau.

* “Handoffs” là một mẫu tương tác phổ biến trong các hệ thống đa tác tử, nơi một tác tử chuyển giao quyền điều khiển cho một tác tử khác.
* Handoffs cho phép xác định:
* destination: tác tử đích để điều hướng tới.
* payload: thông tin để truyền cho tác tử đó.
* Áp dụng cho các kiến trúc Network, Supervisor, Hierarchical.
* Có thể truyền luôn trạng thái toàn cục: lịch sử toàn bộ trò chuyện.
* Tool calls (Gọi công cụ):
* Đây là trường hợp đặc biệt của kiến trúc Supervisor (tool-calling).
* Trong kiến trúc này, các tác tử riêng lẻ có thể được biểu diễn dưới dạng các công cụ.

##### Một số kiến trúc khác

Một số kiến trúc khác như:

* Agent customization
* Multi-Agent Conversations
* Flexible Conversation Patterns



#### Ứng dụng của hệ thống

* Trí tuệ nhân tạo và xử lý ngôn ngữ tự nhiên
* Hệ thống giao thông thông minh
* Thị trường kinh tế
* Chăm sóc khách hàng
* An ninh mạng
* Y tế thông minh
* Khoa học dữ liệu

AI Agent & Multi-Agent Systems: Tạo ra giải pháp thông minh, hợp tác và tối ưu hóa trong thế giới kết nối (API, IoT, Hardware, Software).

#### Ưu điểm của mô hình MAS

* Tính mô-đun (Modularity): Hệ thống dễ mở rộng, bảo trì, và nâng cấp từng phần.
* Tính song song (Parallelism): Các tác tử có thể hoạt động đồng thời, giúp tăng tốc độ xử lý.
* Tính linh hoạt (Flexibility): Có thể điều chỉnh hành vi từng tác tử để thích ứng với yêu cầu mới.
* Khả năng phối hợp (Coordination): Cho phép chia sẻ tri thức và hợp tác giữa các mô hình AI khác nhau (ví dụ: giữa mô hình LLM và mô hình đánh giá điểm số).
* Là công nghệ cốt lõi cho việc thực hiện các nhiệm vụ phân tán (Core technology enabling distributed tasks).
* Chuyên môn hóa giúp đơn giản hóa các nhiệm vụ phức tạp (Specialization simplifies complex tasks).

#### Xu hướng tương lai

Khả năng tương tác: Các tiêu chuẩn mới nổi như MCP và A2A. Tạo điều kiện thuận lợi cho việc giao tiếp giữa các tác nhân và cách sử dụng công cụ.

Dân chủ hóa: Tăng khả năng tiếp cận thông qua các nền tảng không cần mã và ít mã. Giúp phát triển dễ dàng hơn.

Áp dụng doanh nghiệp: Tập trung vào tính tuân thủ, bảo mật và tích hợp. Quan trọng đối với môi trường doanh nghiệp.

Multimodal Agents: Tích hợp giọng nói. Khả năng hình ảnh và văn bản. Nâng cao tương tác giữa các tác nhân.

#### Thách thức

Rủi ro bị khóa chặt bởi nhà cung cấp nếu gắn chặt với hệ sinh thái Google Cloud. Đường cong học tập: Việc điều phối nâng cao và luồng đa tác nhân đòi hỏi phải hiểu các khái niệm trừu tượng ADK.

Chi phí: Việc sử dụng Gemini LLM, lưu trữ và dịch vụ Vertex AI trong môi trường sản xuất sẽ phát sinh chi phí đám mây.

Độ phức tạp của việc gỡ lỗi: Với nhiều tác nhân và các công cụ không đồng bộ, việc theo dõi hành vi cần được thực hiện cẩn thận, may mắn thay, giao diện người dùng (UI) đã hỗ trợ việc này.

#### Ví dụ điển hình

|  |  |  |
| --- | --- | --- |
| **From** | | **Message** |
| User | | Hãy lên kế hoạch du lịch giúp tôi. |
| Travel Agent | | Chắc chắn rồi! Để bắt đầu, bạn có thể cho tôi biết bạn muốn đi đầu và thời gian du lịch của bạn là bao lâu? |
| User | | Tôi muốn đi Đà Nẵng vào cuối tuần này, từ thứ Sáu đến Chủ Nhật. |
| Travel Agent | | Tuyệt vời! Dưới đây là một kế hoạch du lịch cơ bản: - Ngày 1: Đến Đà Nẵng, tham quan Bà Nà Hills - Ngày 2 Khám phá phố cổ Hội An, tham quan chùa Cầu - Ngày 3: Tham quan Ngũ Hành Sơn, và nghỉ ngơi trước khi về. Bạn có muốn chỉnh sửa gì không? |
| User | | Tôi muốn thêm một buổi ăn tối ở nhà hàng ven biển vào ngày thứ 2, và chuyển chuyến bay về sớm hơn một chút. |
| Travel Agent | | Cảm ơn bạn đã cung cấp thông tin! Tôi đã chỉnh sửa lịch trình như sau: - Ngày 1: Đến Đà Nẵng, tham quan Bà Nà Hills - Ngày 2 Khám phá Hội An, ăn tối tại Nhà hàng biển Làng Chài - Ngày 3: Tham quan Ngũ Hành Sơn và bay về vào Th chiều. Bạn đã hài lòng với kế hoạch chưa? |
| User | Rất ổn rồi, nhưng tôi muốn đặt phòng khách sạn 3 sao và cần chuyến bay sớm nhất từ Hà Nội vào Đà Nẵng. |
| Travel Agent | Tuyệt vời Tôi sẽ yêu cầu Agent Khách Sạn và Agent Máy Bay giúp bạn. Chờ một chút nhé. |
| Agent Khách Sạn | Để đáp ứng yêu cầu của bạn, tôi đã tìm được các khách sạn 3 sao ở Đà Nẵng gần biển với giá phải chăng - Khách sạn Sunflower 1 đêm: 500.000 VNĐ - Khách sạn Green Peace – 1 đêm: 450.000 VNĐ Bạn muốn tôi book phòng cho bạn không? |
| User | Chọn Khách sạn Sunflower nhé. |
| Agent Máy Bay | Tôi đã kiểm tra các chuyến bay từ Hà Nội đến Đà Nẵng vào thứ Sáu. Dưới đây là các lựa chọn: - Vietnam Alrlines: Bay lúc 6h sáng, giá 1,200.000 VNĐ - VietJet Bay lúc 7h sáng, giá 1.000.000 VNĐ Bạn muốn chọn chuyến bay nào? |
| User | Chọn Vietnam Airlines lúc 6h sáng. |
| Travel Agent | Đã xong! Tôi đã đặt phòng tại Khách sạn Sunflower và chuyến bay với Vietnam Airlines lúc 6h sáng. Bạn có muốn tôi hoàn tất việc đặt vé tham quan hay chuẩn bị thêm dịch vụ gì nữa không? |

## Các tác tử trong hệ thống Summarization Studio

Trong đề tài này, hệ thống đa tác tử được sử dụng để phân tách quy trình tóm tắt thành các giai đoạn nhỏ, mỗi tác tử đảm nhiệm một nhiệm vụ cụ thể. Ví dụ:

* Agent 1 – Coordinator: Điều phối pipeline, phân công nhiệm vụ cho các agent tương ứng và phản hồi người dùng.
* Agent 2 – Reader/OCR: Đọc và trích xuất nội dung văn bản gốc từ văn bản, ảnh hay file PDF.
* Agent 3 – Spellchecker: Kiểm tra và sửa lỗi chính tả.
* Agent 4 – Extractor: Thực hiện tóm tắt trích xuất sao cho phù hợp với từng khối lớp (1-5).
* Agent 5 – Abstracter: Tạo bản tóm tắt diễn giải tự nhiên bằng mô hình LLM sao cho phù hợp với từng cấp lớp (1-5).
* Agent 6 – Grade Calibrator: Điều chỉnh độ dài, từ vựng sao cho phù hợp với cấp lớp (1-5).
* Agent 7 – Evaluator: Đánh giá chất lượng bản tóm tắt theo tiêu chí độ chính xác, mạch lạc và mức độ phù hợp với học sinh tiểu học, chấm điểm dựa trên độ dễ hiểu theo thang điểm (0-10).
* Agent 8 – Aggregator: Tổng hợp kết quả cuối cùng để xuất bản tóm tắt hoàn chỉnh.

Cách tiếp cận này giúp hệ thống:

Giảm tải cho từng mô hình riêng lẻ nhờ chia nhỏ nhiệm vụ. Tăng độ chính xác và khả năng kiểm soát quy trình, vì từng tác tử có thể được huấn luyện hoặc tinh chỉnh độc lập. Dễ dàng mở rộng hoặc thay thế từng tác tử mà không ảnh hưởng đến toàn bộ hệ thống.

## Công nghệ và nền tảng phát triển

Hệ thống được phát triển dựa trên hai nền tảng chính: **Python** cho mô hình đa tác tử (Multi-Agent System) và **Java Spring Boot** cho phần backend phục vụ giao tiếp giữa hệ thống AI và giao diện người dùng. Và React cho Web và Mobile (đang phát triển).

### Ngôn ngữ và môi trường lập trình

* Python:

Là ngôn ngữ lập trình mạnh mẽ, phổ biến trong lĩnh vực Trí tuệ nhân tạo (AI) và Xử lý ngôn ngữ tự nhiên (NLP). Cung cấp nhiều thư viện hỗ trợ học máy, xử lý dữ liệu, và triển khai mô hình như: transformers, langchain, ollama, chroma, pandas, numpy, v.v. Được sử dụng để xây dựng các tác tử (agents), điều phối hoạt động của từng tác tử và xử lý dữ liệu đầu vào – đầu ra.

* Java Spring Boot:

Là framework mạnh mẽ dùng để xây dựng API backend theo mô hình RESTful. Hỗ trợ khả năng mở rộng, quản lý request và bảo mật dữ liệu giữa hệ thống AI và ứng dụng người dùng. Được sử dụng để kết nối giữa hệ thống MAS (Python) với giao diện web/app, đảm bảo luồng trao đổi dữ liệu mượt mà và ổn định.

### Các công nghệ và thư viện chính

1. Ollama

Nền tảng giúp chạy và quản lý các mô hình ngôn ngữ lớn (LLMs) ngay trên máy cục bộ. Cho phép sử dụng các mô hình như qwen2.5, phi3, llama3, hoặc mô hình tùy chỉnh đã fine-tuned.

Ưu điểm:

* Giảm phụ thuộc vào API bên ngoài (như OpenAI hoặc Gemini).
* Tăng tốc độ phản hồi và đảm bảo tính riêng tư dữ liệu.
* Trong hệ thống, Ollama đóng vai trò mô hình sinh tóm tắt và đánh giá nội dung cho các tác tử như Abstracter và Evaluator.

1. LangChain

Framework chuyên dùng để xây dựng ứng dụng sử dụng LLMs bằng cách kết hợp các thành phần như chuỗi truy vấn (Chains), bộ nhớ (Memory), và công cụ (Tools).

Trong hệ thống, LangChain giúp:

* Tổ chức logic hoạt động của các tác tử.
* Xử lý luồng hội thoại giữa người dùng và các mô hình.
* Tích hợp Ollama và các API khác trong cùng một pipeline xử lý.

1. LangGraph

Là một mở rộng của LangChain, cung cấp kiến trúc dạng đồ thị (graph-based) cho phép quản lý luồng tương tác phức tạp giữa nhiều tác tử (multi-agent orchestration).

Mỗi node trong đồ thị tương ứng với một agent, ví dụ:

|  |
| --- |
| Coordinator Agent → Reader Agent → Spellchecker Agent → Extractor Agent → Abstracter Agent → Grade Calibrator → Evaluator Agent → Aggregator Agent. |

LangGraph giúp đảm bảo:

* Các tác tử hoạt động độc lập nhưng phối hợp nhịp nhàng.
* Dễ dàng giám sát, mở rộng và gỡ lỗi hệ thống.

1. Chroma

Là cơ sở dữ liệu vector (vector database) dùng để lưu trữ và truy vấn dữ liệu dạng embedding (biểu diễn ngữ nghĩa của văn bản).

Trong hệ thống:

* Dùng để lưu trữ embedding của nội dung truyện và bài đọc, giúp tìm kiếm và đối chiếu nhanh giữa các câu hoặc đoạn tương đồng.
* Hỗ trợ cho các tác tử Extractor và Evaluator khi cần truy xuất ngữ cảnh hoặc đánh giá tính nhất quán của tóm tắt.

1. Jupyter Notebook

Là môi trường lập trình tương tác được sử dụng trong giai đoạn nghiên cứu, thử nghiệm và huấn luyện mô hình.

Cho phép:

* Hiển thị kết quả tức thời sau mỗi lần chạy mã.
* Kết hợp giữa code, biểu đồ và ghi chú giải thích, thuận tiện cho quá trình nghiên cứu và trình bày kết quả.

1. Anaconda

Là bộ công cụ hỗ trợ quản lý môi trường Python và cài đặt thư viện khoa học dữ liệu.

Được sử dụng để:

* Tạo môi trường riêng biệt cho dự án MAS (tránh xung đột phiên bản thư viện).
* Quản lý dễ dàng các gói như langchain, chromadb, torch, transformers, v.v.

### Môi trường triển khai

Máy cục bộ (Localhost): Toàn bộ hệ thống MAS chạy trên máy tính cá nhân thông qua Ollama và Python environment kernel.

Server backend (Spring Boot): Được triển khai trên cổng riêng (ví dụ: localhost:8080), đóng vai trò kết nối và quản lý API.

Giao diện người dùng: Có thể được xây dựng dưới dạng ứng dụng web hoặc tích hợp với dashboard để hiển thị kết quả tóm tắt, đánh giá và bài học rút ra (đang phát triển).

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG TÓM TẮT TRUYỆN, BÀI ĐỌC

## Phân tích yêu cầu hệ thống

### Mục tiêu hệ thống

### Phân tích yêu cầu chức năng

## Kiến trúc hệ thống

### Thiết kế hệ thống đa tác tử

### Thiết kế cơ sở dữ liệu

### Thiết kế API và tích hợp

## Thu thập và xử lý dữ liệu



CHƯƠNG 4: XÂY DỰNG VÀ TRIỂN KHAI HỆ THỐNG TÓM TẮT

## Môi trường và công cụ phát triển

## Xây dựng pipeline đa tác tử (Multi-Agent Implementation)

## Xây dựng hệ thống mobile và web

### Xây dựng API và thiết kế giao diện

## Thiết lập cơ sở dữ liệu

## Triển khai hệ thống

## Kết quả, kiểm thử và đánh giá

## 

## Xây dựng API và thiết kế giao diện

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN



## Đánh giá kết quả đạt được

## Hạn chế và thách thức

## Hạn chế của nghiên cứu

## Hướng phát triển tương lai

KẾT LUẬN

Em xin gửi lời biết ơn sâu sắc đến thầy Hoàng Ngọc Long đã dành nhiều thời gian và tâm huyết hướng dẫn nghiên cứu và giúp em hoàn thành môn học.

Em đã có nhiều cố gắng hoàn thiện dự án bằng tất cả năng lực của mình, tuy nhiên không thể tránh khỏi nhiều thiếu sót, rất mong nhận được những đóng góp quý báu của quý thầy cô và các bạn.