ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: AN TOÀN THÔNG TIN

ĐỀ TÀI:

XÂY DỰNG ỨNG DỤNG TÓM TẮT VĂN BẢN Y TẾ

Người hướng dẫn: **TS. HUỲNH HỮU HƯNG**

Sinh viên thực hiện:

Họ và tên	Lớp	Mã số sinh viên
KIỀU DƯƠNG TÂY	**21TCLC_DT3**	**102210231**
NGUYỄN XUÂN THỊNH	**21TCLC_DT3**	**102210234**

Đà Nẵng, năm 2026

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: AN TOÀN THÔNG TIN

ĐỀ TÀI:

XÂY DỰNG ỨNG DỤNG TÓM TẮT VĂN BẢN Y TẾ

Người hướng dẫn: **TS. HUỲNH HỮU HƯNG**
Người duyệt:

Sinh viên thực hiện:

Họ và tên	Lớp	Mã số sinh viên
KIỀU DƯƠNG TÂY	**21TCLC_DT3**	**102210231**
NGUYỄN XUÂN THỊNH	**21TCLC_DT3**	**102210234**

Đà Nẵng, năm 2026

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

NHẬN XÉT ĐỒ ÁN TỐT NGHIỆP

I. Thông tin chung:

1. Họ, tên sinh viên: Kiều Dương Tây - Nguyễn Xuân Thịnh
Lớp: 21TCLC_DT3 Số thẻ SV: 102210231-102210234
2. Tên đề tài: Xây dựng ứng dụng tóm tắt văn bản y tế
3. Người hướng dẫn: TS. Huỳnh Hữu Hưng Học hàm/ học vị: Tiến Sĩ

II. Nhận xét, đánh giá đồ án tốt nghiệp:

1. Về tính cấp thiết, tính mới, khả năng ứng dụng của đề tài: (điểm tối đa là 2đ)

……..
……..

2. Về kết quả giải quyết các nội dung nhiệm vụ yêu cầu của đồ án: (điểm tối đa là 4đ)

……..
……..

3. Về hình thức, cấu trúc, bố cục của đồ án tốt nghiệp: (điểm tối đa là 2đ)

……..
……..

4. Đề tài có giá trị khoa học/ có bài báo/ giải quyết vấn đề đặt ra của doanh nghiệp hoặc nhà trường: (điểm tối đa là 1đ)

……..
……..

5. Các tồn tại, thiếu sót cần bổ sung, chỉnh sửa:

……..
……..

III. Tinh thần, thái độ làm việc của sinh viên: (điểm tối đa 1đ)

……..

IV. Đánh giá:

6. Điểm đánh giá: ……../10 (lấy đến 1 số lẻ thập phân)
7. Đề nghị: ☐ Được bảo vệ đồ án ☐ Bổ sung để bảo vệ ☐ Không được bảo vệ

Đà Nẵng, ngày tháng năm 2025

Trưởng Bộ môn **Người hướng dẫn**

TÓM TẮT

Tên đề tài: **Xây dựng ứng dụng tóm tắt văn bản y tế**

Sinh viên thực hiện:	Số thẻ SV:	Lớp:
Kiều Dương Tây	102210231	21TCLC_DT3
Nguyễn Xuân Thịnh	102210234	21TCLC_DT3

Xây dựng ứng dụng di động **tóm tắt văn bản y tế** là một đề tài hướng đến việc hỗ trợ người dùng tiếp cận nhanh chóng và hiệu quả với các tài liệu chuyên ngành trong lĩnh vực y học. Trong bối cảnh ngành y tế tạo ra một khối lượng lớn văn bản mỗi ngày như bài báo nghiên cứu, hồ sơ bệnh án, tài liệu chuyên môn và báo cáo lâm sàng, việc đọc và xử lý thủ công toàn bộ nội dung này đòi hỏi nhiều thời gian và công sức.

Ứng dụng được đề xuất cho phép người dùng nhập văn bản hoặc tải lên tài liệu y tế, sau đó hệ thống sẽ tự động phân tích và tạo ra bản tóm tắt ngắn gọn, diễn đạt lại khái quát nội dung. Qua đó, ứng dụng giúp người dùng nhanh chóng nắm bắt ý chính của tài liệu, hỗ trợ hiệu quả cho việc học tập, nghiên cứu và tham khảo thông tin y khoa.

Đề tài áp dụng các kỹ thuật Trí tuệ nhân tạo và Xử lý ngôn ngữ tự nhiên (NLP), đặc biệt là các mô hình ngôn ngữ lớn dựa trên kiến trúc Transformer như T5, BART, được tinh chỉnh (fine-tuning) cho dữ liệu y tế tiếng Anh.

Đề xuất kỹ thuật được áp dụng cho dự án bao gồm:

- Ngôn ngữ lập trình: Kotlin (Android), SwiftUI (iOS)
- Không gian/Công cụ phát triển chính: Android Studio, Xcode, Cursor IDE
- Công cụ: FastAPI
- AI: Tóm tắt văn bản, xử lý ngôn ngữ tự nhiên cho chatbot
- Framework/Thư viện: Android UI, SwiftUI

Với mục tiêu đặt ra là thiết kế và triển khai một ứng dụng di động có khả năng tóm tắt hiệu quả văn bản y tế chuyên biệt, đề tài tập trung giải quyết bài toán xử lý ngôn ngữ chuyên ngành, tối ưu hóa mô hình AI cho môi trường di động và đánh giá tính khả thi của việc chạy các mô hình Transformer sau khi nén trên phần cứng hạn chế.

Tổng kết lại, ứng dụng tóm tắt văn bản y tế mang lại nhiều lợi ích thiết thực như tiết kiệm thời gian đọc tài liệu, nâng cao hiệu quả học tập và nghiên cứu, hỗ trợ người dùng tiếp cận nhanh thông tin y khoa quan trọng, đồng thời mở ra tiềm năng phát triển các ứng dụng AI chuyên sâu trong lĩnh vực chăm sóc sức khỏe và giáo dục y học

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên:

Kiều Dương Tây Lớp: 21TCLC_DT3 Mã số sinh viên: 102210231

Nguyễn Xuân Thịnh Lớp: 21TCLC_DT3 Mã số sinh viên: 102210234

Khoa: Công nghệ thông tin Ngành: An toàn thông tin

1. *Tên đề tài đồ án:* Xây dựng ứng dụng tóm tắt văn bản y tế
2. *Đề tài thuộc diện:* ☐ *Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện*
3. *Các số liệu và dữ liệu ban đầu:* Không có
4. *Nội dung các phần thuyết minh và tính toán:*

- Mở đầu: Giới thiệu về nhu cầu thực tế và lý do thực hiện đề tài, đồng thời giới thiệu mục đích, mục tiêu hướng đến, tính năng và đối tượng.

- Chương 1: Cơ sở lý thuyết: Giới thiệu tổng quan về các công nghệ và mô hình được sử dụng trong dự án.

- Chương 2: Phân tích và thiết kế hệ thống: Trình bày thiết hệ thống, hướng tiếp cận vấn đề.

- Chương 3: Triển khai và đánh giá: Trình bày cách cài đặt, vận hành. Trình bày những kết quả và đánh giá thu được

- Kết luận và hướng phát triển: Trình bày các kết quả đạt được, chỉ ra những hạn chế còn tồn tại và đề xuất hướng phát triển.

5. *Các bản vẽ, đồ thị:* Không có
6. *Họ tên người hướng dẫn:* TS. Huỳnh Hữu Hưng
7. *Ngày giao nhiệm vụ đồ án:* *……/……/2025*
8. *Ngày hoàn thành đồ án:* *……/……/2026*

Đà Nẵng, ngày tháng năm 2026

Trưởng Bộ môn Người hướng dẫn

LỜI NÓI ĐẦU

Trong lời đầu tiên của báo cáo đồ án tốt nghiệp "Xây dựng ứng dụng tóm tắt văn bản y tế", chúng em xin được bày tỏ lòng biết ơn chân thành đến tất cả những người đã luôn đồng hành, hỗ trợ và tạo điều kiện thuận lợi cho chúng em trong suốt quá trình học tập và thực hiện đồ án.

Trước hết, chúng em xin gửi lời cảm ơn sâu sắc đến thầy **TS. Huỳnh Hữu Hưng**, giảng viên đã tận tâm hướng dẫn chúng em trong suốt quá trình thực hiện đề tài. Nhờ sự hướng dẫn, hỗ trợ và những góp ý quý báu, chi tiết từ Thầy trong từng buổi báo cáo tiến độ, chúng em đã có thể hoàn thiện đề tài đồ án này. Trong suốt quá trình thực hiện đề tài, chúng em đã có cơ hội tiếp cận thực tế với quy trình phát triển phần mềm, rèn luyện kỹ năng tư duy hệ thống, tổ chức dữ liệu và thiết kế giao diện người dùng, đồng thời nâng cao khả năng vận dụng kiến thức về trí tuệ nhân tạo trong lĩnh vực ứng dụng di động.

Chúng em cũng xin chân thành cảm ơn **Ban Giám hiệu nhà trường**, cùng toàn thể quý thầy cô trong **Khoa Công nghệ Thông tin – Trường Đại học Bách khoa – Đại học Đà Nẵng,** những người đã truyền đạt kiến thức và kinh nghiệm quý báu trong suốt quá trình học tập ngành công nghệ thông tin. Chính nhờ những nền tảng kiến thức vững chắc đó, chúng em đã có đủ hành trang để nghiên cứu, học tập và thực hiện đề tài một cách hiệu quả.

Do thời gian thực hiện có hạn và bản thân còn nhiều thiếu sót về kinh nghiệm thực tiễn, đồ án chắc chắn không thể tránh khỏi những hạn chế nhất định. Chúng em rất mong nhận được phản hồi tích cực và góp ý chân thành từ phía thầy cô cũng như hội đồng chấm thi.

Chúng em xin chân thành cảm ơn!

CAM ĐOAN

Tôi xin cam đoan:

Báo cáo đồ án tốt nghiệp với tên đề tài "Xây dựng ứng dụng tóm tắt văn bản y tế" là công trình nghiên cứu của chính sinh viên Kiều Dương Tây và Nguyễn Xuân Thịnh dưới sự hướng dẫn trực tiếp của giảng viên TS. Huỳnh Hữu Hưng

1. Tôi đã tự đọc nghiên cứu, dịch tài liệu và tổng hợp các kiến thức đã làm nên báo cáo này và đảm bảo không sao chép ở bất cứ đâu.

2. Những lý thuyết trong luận văn đều được sử dụng tài liệu như tôi đã tham khảo ở phần tài liệu tham khảo đã có trong báo cáo.

Nếu có vi phạm, tôi xin chịu hoàn toàn trách nhiệm.

Đà Nẵng, ngày tháng năm 2026

Sinh viên thực hiện

Kiều Dương Tây Nguyễn Xuân Thịnh

MỤC LỤC

DANH SÁCH HÌNH VẼ

DANH SÁCH BẢNG BIỂU

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

Chữ viết tắt	Tên đầy đủ
CNTT	Công Nghệ Thông Tin
API	Application Programming Interface
XML	eXtensible Markup Language
UI	User Interface
IDE	Integrated Development Environment
NLP	Natural Language Processing
JSON	JavaScript Object Notation
RESTful	Representational State Transfer
SDK	Software Development Kit
PDF	Portable Document Format
DOC	Document
DOCX	Document Open XML
BTS	Biomedical Text Summarization
PEFT	Parameter Efficient Fine-Tuning
LoRA	Low-Rank Adaptation
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
NER	Named Entity Recognition

MỞ ĐẦU

1. Giới thiệu đề tài

Xuất phát từ nhu cầu thực tế hiện nay, lượng thông tin và tài liệu văn bản ngày càng gia tăng mạnh mẽ trong nhiều lĩnh vực như giáo dục, y tế, thể thao, báo chí và nghiên cứu khoa học. Người dùng thường xuyên phải tiếp cận các tài liệu có độ dài lớn như bài báo, giáo trình, tài liệu nghiên cứu, báo cáo chuyên ngành hoặc hồ sơ PDF, DOC. Việc đọc và nắm bắt toàn bộ nội dung trong thời gian ngắn là một thách thức lớn, đặc biệt đối với sinh viên và người đi làm có quỹ thời gian hạn chế.

Trong bối cảnh đó, sự phát triển của trí tuệ nhân tạo (AI), đặc biệt là xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP), đã mở ra nhiều hướng tiếp cận hiệu quả trong việc tự động phân tích và tóm tắt nội dung văn bản. Các mô hình AI hiện đại có khả năng trích xuất ý chính, làm nổi bật thông tin quan trọng, diễn đạt lại nội dung và thậm chí tạo câu hỏi dựa trên văn bản gốc, giúp người dùng hiểu nhanh và ghi nhớ tốt hơn.

Từ nhu cầu thực tiễn trong lĩnh vực y tế, đề tài "**Xây dựng ứng dụng tóm tắt văn bản y tế**" được thực hiện nhằm phát triển một hệ thống thông minh cho phép người dùng nhập các tài liệu y khoa như bài báo khoa học, hướng dẫn điều trị hoặc văn bản chuyên ngành dưới dạng PDF, DOC. Hệ thống tự động phân tích nội dung và tạo ra bản tóm tắt ngắn gọn, tập trung vào các thông tin y học cốt lõi, giúp người dùng nhanh chóng nắm bắt nội dung chính của tài liệu. Ứng dụng hướng đến mục tiêu hỗ trợ sinh viên y, nhân viên y tế và nhà nghiên cứu tiếp cận hiệu quả khối lượng lớn tài liệu chuyên môn, từ đó giảm thời gian đọc hiểu và xử lý thông tin, nâng cao hiệu quả học tập, nghiên cứu và công việc chuyên môn.

2. Mục đích thực hiện đề tài

- Thiết kế và xây dựng một ứng dụng di động hỗ trợ phân tích và tóm tắt tài liệu văn bản dựa trên trí tuệ nhân tạo.
- Ứng dụng các kiến thức đã học về xử lý ngôn ngữ tự nhiên, lập trình ứng dụng di động và thiết kế hệ thống backend để giải quyết một bài toán thực tế.
- Cung cấp cho người dùng một công cụ giúp hiểu nhanh nội dung tài liệu dài mà không cần đọc toàn bộ văn bản gốc.

3. Mục tiêu hướng đến

- Phát triển hệ thống có khả năng tiếp nhận đầu vào là văn bản thuần hoặc file tài liệu (PDF, DOC/DOCX).
- Tự động tóm tắt nội dung chính của tài liệu bằng mô hình AI.
- Tích hợp chatbot tư vấn và đề xuất trang phục sử dụng xử lý ngôn ngữ tự nhiên

4. Đối tượng người dùng

- Sinh viên và học viên trong lĩnh vực y – sinh học, có nhu cầu đọc, ôn tập giáo trình, tài liệu chuyên ngành và bài báo khoa học.

- Nhân viên y tế và nhà nghiên cứu, thường xuyên tiếp cận các tài liệu y khoa dài như hướng dẫn điều trị, báo cáo nghiên cứu, tài liệu chuyên môn.

- Người quan tâm đến việc ứng dụng trí tuệ nhân tạo trong hỗ trợ tiếp cận và xử lý tài liệu y tế, nhằm nâng cao hiệu quả học tập, nghiên cứu và công việc chuyên môn.

5. Phạm vi và đối tượng

- Phạm vi: Ứng dụng được triển khai thử nghiệm trên nền tảng Android và iOS với các chức năng cơ bản gồm nhập văn bản/tài liệu, phân tích nội dung và trả về kết quả tóm tắt, tư vấn bằng chatbot

- Đối tượng sử dụng: Người dùng cá nhân sử dụng điện thoại thông minh, có nhu cầu đọc và xử lý tài liệu văn bản thường xuyên.

6. Công nghệ phát triển

- Frontend Android: Kotlin
- Frontend iOS: SwiftUI
- Backend: FastAPI (Python)
- Cơ sở dữ liệu: Room, CoreData
- AI: Gemini AI
- Quản lý và lưu trữ dự án: Github

7. Cấu trúc đồ án tốt nghiệp

- Mở đầu
- Chương 1: Cơ sở lý thuyết
- Chương 2: Phân tích và thiết kế hệ thống
- Chương 3: Triển khai và đánh giá
- Kết luận và hướng phát triển

8. Phân chia nội dung công việc

Công việc	Người thực hiện
Viết đề cương, mô tả mục tiêu và tính cấp thiết của đề tài	Kiều Dương Tây
Thiết kế giao diện trên figma	Kiều Dương Tây
Phân tích nghiệp vụ, vẽ sơ đồ Use Case, sơ đồ tuần tự	Cả nhóm
Triển khai giao diện iOS	Nguyễn Xuân Thịnh
Triển khai giao diện Android	Kiều Dương Tây

Công việc	Người thực hiện
Viết API cho toàn bộ hệ thống	Cả nhóm
Tích hợp Gemini Chatbot	Nguyễn Xuân Thịnh
Xây dựng chức năng xử lý văn bản	Cả nhóm
Thiết kế và triển khai hệ thống backend với Fast API	Cả nhóm
Kiểm thử chức năng ứng dụng trên cả Android và iOS	Nguyễn Xuân Thịnh
Viết tài liệu báo cáo và chuẩn bị thuyết trình	Kiều Dương Tây

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. Tổng quan về trí tuệ nhân tạo trong xử lý ngôn ngữ tự nhiên

1.1.1. Tiền xử lý văn bản

- Trong xử lý ngôn ngữ tự nhiên (NLP) áp dụng cho lĩnh vực y tế, bước tiền xử lý văn bản là yếu tố then chốt để chuẩn hóa dữ liệu đầu vào, giảm nhiễu từ cấu trúc tài liệu lâm sàng phức tạp và nâng cao hiệu suất của các mô hình học sâu sau đó [1].
- Trong đồ án "Xây dựng ứng dụng tóm tắt văn bản y tế", hệ thống nhận đầu vào chủ yếu là văn bản tiếng Anh (từ tài liệu PDF/DOC/DOCX như báo cáo bệnh án, tóm tắt xuất viện, bài báo khoa học PubMed, hoặc hướng dẫn điều trị). Do input là tiếng Anh, quy trình tiền xử lý được tối ưu hóa để xử lý đặc thù của văn bản y tế tiếng Anh: thuật ngữ chuyên môn dày đặc, viết tắt (e.g., HTN for hypertension, MI for myocardial infarction), entity y khoa (tên thuốc, mã ICD-10), và cấu trúc không đồng nhất [2].

Quy trình tiền xử lý bao gồm các bước chính:

- **Trích xuất nội dung văn bản:** Tài liệu PDF và DOC/DOCX được chuyển đổi sang văn bản thuần bằng thư viện chuyên dụng (ví dụ: PyMuPDF cho PDF, python-docx cho DOCX). Bước này loại bỏ header/footer, bảng biểu, hình ảnh không cần thiết – rất quan trọng với báo cáo y tế có định dạng đa dạng và thường chứa bảng kết quả xét nghiệm [1].
- **Chuẩn hóa văn bản:** Chuyển toàn bộ về mã Unicode UTF-8, loại bỏ ký tự đặc biệt thừa, khoảng trắng dư thừa, và chuẩn hóa một phần viết tắt phổ biến (nếu có từ điển tùy chỉnh). Tuy nhiên, các entity y khoa quan trọng (tên thuốc như "aspirin", viết tắt "HbA1c", mã bệnh "ICD-10: E11") được giữ nguyên để tránh mất thông tin ngữ nghĩa, giảm nguy cơ lỗi khi tóm tắt [3].

Những bước này đặc biệt hiệu quả khi input là tiếng Anh, giúp mô hình Transformer nắm bắt tốt hơn ngữ cảnh y tế dài và phức tạp, đồng thời giảm hallucination trong tóm tắt [1].

1.1.2. Phân tích và tóm tắt nội dung bằng mô hình AI

Sau bước tiền xử lý, hệ thống sử dụng các mô hình trí tuệ nhân tạo dựa trên kiến trúc Transformer để phân tích nội dung và thực hiện tóm tắt văn bản [4].

Trong đồ án này, mô hình được chọn là LongT5-tglobal-base (biến thể long-t5-tglobal-base-16384-booksum) – một phiên bản mở rộng của T5 hỗ trợ xử lý chuỗi đầu vào dài lên đến 16.384 tokens nhờ cơ chế attention toàn cục-cục bộ (global-local attention), giúp khắc phục hạn chế về độ dài ngữ cảnh của T5 gốc [7], [8]. Mô hình này

được pre-train và fine-tune ban đầu trên dataset BookSum – tập dữ liệu tóm tắt sách dài – nên rất phù hợp cho nhiệm vụ tóm tắt abstractive các tài liệu dài, phức tạp như báo cáo lâm sàng hoặc bài báo y khoa tiếng Anh [7].

Chức năng Tóm tắt văn bản (Text Summarization): Mô hình thực hiện tóm tắt abstractive, tự động rút gọn tài liệu y tế tiếng Anh dài (báo cáo bệnh án, hồ sơ xuất viện, hướng dẫn điều trị, tài liệu PubMed) bằng cách giữ lại các ý chính bao gồm triệu chứng, chẩn đoán, phương pháp điều trị, kết quả lâm sàng và khuyến nghị. Việc sử dụng LongT5 giúp duy trì ngữ cảnh toàn bộ tài liệu mà không cần cắt đoạn, giảm nguy cơ mất thông tin quan trọng trong văn bản y tế [1], [9].

Để thích nghi với lĩnh vực y tế, mô hình đã được fine-tune trên tập dữ liệu y tế chuyên biệt. Quá trình fine-tune tập trung vào việc cải thiện độ chính xác entity y khoa (tên thuốc, mã bệnh, viết tắt), giảm hallucination (tạo thông tin sai lệch) và tăng tính faithful (trung thực với nguồn) – các thách thức phổ biến trong biomedical text summarization [1], [2], [10].

So với các mô hình summarization khác (như BART hoặc PEGASUS), LongT5 nổi bật ở khả năng xử lý input dài mà không cần hierarchical segmentation phức tạp, phù hợp với đặc thù tài liệu y tế tiếng Anh thường có độ dài lớn và cấu trúc không đồng nhất [7], [11]. Hướng tiếp cận này trong đồ án không chỉ tận dụng pre-trained knowledge từ BookSum mà còn domain adaptation qua fine-tune, góp phần nâng cao hiệu suất tóm tắt trong bối cảnh y tế Việt Nam (nơi tài liệu chuyên sâu thường tham chiếu nguồn tiếng Anh).

1.2. Tổng quan về chatbot và xử lý ngôn ngữ tự nhiên (NLP):

1.2.1. Khái niệm và lịch sử phát triển của chatbot

Chatbot là hệ thống phần mềm mô phỏng cuộc hội thoại với con người qua văn bản hoặc giọng nói, nhằm cung cấp thông tin, hỗ trợ tác vụ hoặc tư vấn cơ bản. Các chatbot hiện đại dựa trên trí tuệ nhân tạo (AI) và xử lý ngôn ngữ tự nhiên (NLP) để hiểu ý định người dùng, phân tích ngữ nghĩa và tạo phản hồi tự nhiên, chính xác hơn so với các hệ thống dựa trên quy tắc [1].

Lịch sử phát triển chatbot có thể chia thành các giai đoạn chính:

- 1966: ELIZA – chatbot đầu tiên do Joseph Weizenbaum phát triển tại MIT, sử dụng kỹ thuật khớp từ khóa (pattern matching) để mô phỏng vai trò nhà trị liệu tâm lý, chứng minh khả năng tạo ảo giác hội thoại thông minh dù không hiểu sâu [2].
- 1995: ALICE (Artificial Linguistic Internet Computer Entity) – sử dụng ngôn ngữ AIML (Artificial Intelligence Markup Language), cho phép xây dựng quy tắc hội thoại linh hoạt hơn và trở thành nền tảng cho nhiều chatbot mã nguồn mở sau này [2].

- 2011: Siri của Apple – trợ lý ảo đầu tiên tích hợp sâu NLP và nhận diện giọng nói trên thiết bị di động thương mại, mở đường cho các trợ lý ảo cá nhân hóa [1].
- Từ 2020: Sự bùng nổ của các mô hình ngôn ngữ lớn (LLM) như GPT-3/GPT-4, cho phép chatbot xử lý ngữ cảnh dài, sáng tạo nội dung và hội thoại linh hoạt, dẫn đến sự phát triển mạnh mẽ của chatbot dựa trên generative AI trong các lĩnh vực như y tế [3].

Trong bối cảnh y tế, lịch sử này cho thấy sự chuyển dịch từ chatbot quy tắc đơn giản sang LLM-based, giúp chatbot tư vấn sức khỏe trở nên thông minh hơn, nhưng vẫn cần giới hạn để tránh thông tin sai lệch [4]

1.2.2. Tích hợp chatbot tư vấn sức khoẻ sử dụng Gemini API

Chatbot đóng vai trò là trợ lý tư vấn sức khỏe ảo, hỗ trợ người dùng tra cứu và tiếp cận thông tin y tế cơ bản dựa trên các câu hỏi liên quan đến triệu chứng, thói quen sinh hoạt, chăm sóc sức khỏe và kiến thức y học phổ thông. Chatbot được xây dựng trên nền tảng Gemini API, sử dụng mô hình ngôn ngữ lớn để xử lý ngôn ngữ tự nhiên và sinh câu trả lời phù hợp với ngữ cảnh y tế, giúp người dùng tiếp cận thông tin một cách thuận tiện và dễ hiểu.

- Quy trình hoạt động của chatbot
    - Người dùng đặt câu hỏi: Ví dụ: “Tôi thường xuyên mất ngủ, điều này có ảnh hưởng gì đến sức khỏe không?”
    - Gửi câu hỏi lên server: Ứng dụng di động gửi câu hỏi đến backend, nơi tiếp nhận và xử lý yêu cầu thông qua dịch vụ ChatbotService
    - Gọi Gemini API: Câu hỏi được kết hợp với prompt định hướng lĩnh vực sức khỏe, nhằm đảm bảo nội dung phản hồi mang tính tham khảo, không thay thế tư vấn y khoa chuyên môn
    - Xử lý và sinh câu trả lời: Mô hình Gemini phân tích nội dung câu hỏi, ngữ cảnh sức khỏe liên quan và tạo phản hồi phù hợp, dễ hiểu cho người dùng.
    - Trả về client: Câu trả lời được gửi lại ứng dụng và hiển thị dưới dạng hội thoại tự nhiên.
- Ưu điểm khi sử dụng Gemini API trong tư vấn sức khỏe:
    - Khả năng hiểu ngữ cảnh tốt: Gemini có khả năng phân tích câu hỏi sức khỏe ở nhiều mức độ khác nhau, giúp đưa ra phản hồi phù hợp với ngữ cảnh.
    - Hỗ trợ cá nhân hóa thông tin: Chatbot có thể điều chỉnh nội dung tư vấn dựa trên thông tin người dùng cung cấp (ở mức tham khảo).
    - Tương tác tự nhiên: Giao diện hội thoại thân thiện, giúp người dùng dễ dàng tiếp cận các kiến thức chăm sóc sức khỏe cơ bản.

- Khả năng mở rộng: Có thể tích hợp thêm các chức năng như gợi ý lối sống lành mạnh, nhắc nhở chăm sóc sức khỏe hoặc kết hợp với hệ thống phân tích văn bản y tế.

1.2.3. Chatbot tư vấn sức khoẻ sử dụng NLP

Xử lý ngôn ngữ tự nhiên (NLP) là thành phần cốt lõi giúp chatbot hiểu và phản hồi chính xác các câu hỏi liên quan đến tài liệu y tế. NLP hỗ trợ phân tích ý định (intent detection), trích xuất thực thể (entity extraction như tên bệnh, triệu chứng) và nắm bắt ngữ cảnh để sinh phản hồi phù hợp [8].

Trong đồ án này, chatbot dựa hoàn toàn vào mô hình ngôn ngữ lớn (LLM) của Gemini, thay thế các pipeline NLP truyền thống (như rule-based hoặc BERT fine-tune), để xử lý toàn bộ chuỗi: phân tích cú pháp, hiểu ngữ nghĩa, liên kết với tài liệu y tế đầu vào và sinh câu trả lời. Chatbot chỉ hỗ trợ giải thích, tóm tắt hoặc làm rõ thông tin từ nguồn tài liệu đáng tin cậy, không thực hiện chẩn đoán y khoa [4], [9]

Ví dụ phân tích câu hỏi: "Những triệu chứng thường gặp của bệnh tiểu đường type 2 là gì?"

- Intent: Tra cứu / giải thích thông tin y tế
- Entity: Bệnh lý: Tiểu đường type 2, Yêu cầu thông tin: Triệu chứng
- Ngữ cảnh: Dựa trên nội dung tài liệu y tế đã được hệ thống phân tích trước đó
- Output mong muốn: Danh sách các triệu chứng phổ biến (ví dụ: khát nước nhiều, mệt mỏi, tiểu nhiều lần…), được trình bày ngắn gọn, dễ hiểu và bám sát nội dung tài liệu nguồn

Việc sử dụng LLM như Gemini giúp đơn giản hóa quy trình NLP, tăng tốc độ và độ chính xác trong tư vấn sức khỏe, nhưng đòi hỏi đánh giá liên tục để đảm bảo tính faithful và an toàn [3], [4].

1.3. Tổng quan về tóm tắt văn bản y tế (Biomedical Text Summarization - BTS)

Tóm tắt văn bản y tế (Biomedical Text Summarization - BTS) là một nhiệm vụ cốt lõi trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) áp dụng cho y sinh học, nhằm rút gọn các tài liệu y tế dài và phức tạp thành phiên bản ngắn gọn hơn, giữ nguyên các thông tin quan trọng nhất mà vẫn đảm bảo tính chính xác và trung thực về ngữ nghĩa. Với sự bùng nổ dữ liệu y tế không cấu trúc – PubMed hiện có hơn 35 triệu bài báo khoa học, báo cáo lâm sàng, hồ sơ bệnh án điện tử (EHR), hướng dẫn điều trị và tài liệu nghiên cứu – BTS trở thành công cụ thiết yếu giúp bác sĩ, nhà nghiên cứu, dược sĩ và cả bệnh nhân tiếp cận thông tin nhanh chóng, hỗ trợ ra quyết định dựa trên bằng

chứng (evidence-based medicine) và giảm tải công việc đọc tài liệu dài dòng [1], [12].

Khác với tóm tắt văn bản thông thường (general text summarization) BTS phải đối mặt với đặc thù của ngôn ngữ y tế: thuật ngữ chuyên môn dày đặc, cấu trúc câu phức tạp, entity y khoa (tên thuốc, mã bệnh ICD-10, tên xét nghiệm, triệu chứng), viết tắt phổ biến (ví dụ: HTN cho hypertension, MI cho myocardial infarction, HbA1c cho hemoglobin A1c), và tính nhạy cảm cao về độ chính xác (một lỗi nhỏ có thể dẫn đến hiểu lầm nghiêm trọng về chẩn đoán hoặc điều trị). Do đó, BTS không chỉ tập trung vào độ ngắn gọn mà còn phải ưu tiên tính faithful (trung thực với nguồn), giảm hallucination (tạo thông tin sai lệch), và bảo toàn ngữ nghĩa chuyên ngành [13], [14].

Các nghiên cứu gần đây cho thấy BTS đã chuyển dịch mạnh mẽ từ các phương pháp truyền thống (dựa trên quy tắc hoặc học máy cổ điển) sang các mô hình học sâu dựa trên Transformer và mô hình ngôn ngữ lớn (LLM), giúp cải thiện đáng kể chất lượng tóm tắt abstractive. Tuy nhiên, lĩnh vực này vẫn đang phát triển nhanh chóng nhờ sự tích hợp của các kỹ thuật như parameter-efficient fine-tuning (PEFT) và long-document handling, đặc biệt phù hợp với tài liệu y tế dài như bài báo PubMed [1], [15].

1.3.1. Phân loại tóm tắt văn bản y tế

Các phương pháp BTS được phân loại dựa trên cách tiếp cận sinh tóm tắt, bao gồm ba loại chính: extractive, abstractive và hybrid.

- Tóm tắt trích xuất (**Extractive Summarization**): Phương pháp này chọn và ghép các câu hoặc đoạn quan trọng trực tiếp từ văn bản nguồn mà không thay đổi nội dung gốc. Ưu điểm lớn nhất là độ trung thực cao (không hallucination), dễ kiểm soát và dễ giải thích (traceable back to source). Tuy nhiên, tóm tắt có thể thiếu mạch lạc, lặp ý hoặc không diễn đạt mượt mà.

  Ví dụ cụ thể trong y tế: Từ một bài báo PubMed về nghiên cứu thuốc điều trị COVID-19 dài 5000 từ, hệ thống extractive có thể chọn các câu như: “The trial enrolled 1200 patients with moderate-to-severe COVID-19. Remdesivir reduced recovery time by 4 days compared to placebo ($p<0.001$).” và ghép thành tóm tắt ngắn. Phương pháp này phổ biến trong các hệ thống early warning hoặc tóm tắt báo cáo lâm sàng nhanh, nơi độ chính xác tuyệt đối là ưu tiên [13], [16].

- Tóm tắt trừu tượng (**Abstractive Summarization**): Phương pháp này sử dụng mô hình học sâu (thường dựa trên Transformer) để hiểu ý nghĩa toàn bộ văn bản và sinh ra câu mới, diễn đạt lại bằng từ ngữ khác nhưng giữ nguyên ý chính. Kết quả tự nhiên hơn, ngắn gọn hơn và có thể tổng hợp thông tin từ nhiều phần. Tuy nhiên, dễ gặp

hallucination (tạo thông tin không có trong nguồn) và lỗi entity (ví dụ: thay đổi liều lượng thuốc hoặc nhầm mã bệnh).

Ví dụ cụ thể trong y tế: Từ cùng bài báo về Remdesivir, mô hình abstractive có thể sinh ra: "Nghiên cứu lâm sàng cho thấy Remdesivir rút ngắn thời gian hồi phục ở bệnh nhân COVID-19 trung bình đến nặng khoảng 4 ngày so với nhóm placebo, với ý nghĩa thống kê cao." – câu này ngắn hơn, mạch lạc hơn nhưng có nguy cơ thêm chi tiết không chính xác nếu mô hình không được fine-tune tốt [1], [14].

- Tóm tắt lai (**Hybrid Summarization**): Kết hợp ưu điểm của hai loại trên: đầu tiên extractive để chọn các câu chính xác, sau đó abstractive để diễn đạt lại và làm mượt mà. Phương pháp này ngày càng phổ biến trong y tế nhờ cân bằng giữa faithfulness và readability.

  Ví dụ cụ thể: Trong hệ thống tóm tắt hồ sơ bệnh án (EHR), hybrid có thể extractive lấy các câu chẩn đoán chính ("Patient diagnosed with type 2 diabetes mellitus, HbA1c 8.5%"), rồi abstractive tổng hợp thành: "Bệnh nhân được chẩn đoán tiểu đường type 2 với mức HbA1c 8.5%, cho thấy kiểm soát đường huyết kém." – giúp bác sĩ đọc nhanh mà vẫn giữ nguyên entity quan trọng [12], [17].

1.3.2. Thách thức chính trong BTS

BTS gặp nhiều thách thức đặc thù so với tóm tắt văn bản thông thường, đòi hỏi các giải pháp kỹ thuật tiên tiến:

- Độ dài tài liệu dài (Long-document Challenge): Tài liệu y tế như bài báo PubMed hoặc báo cáo lâm sàng thường vượt 4.000–15.000 tokens, vượt giới hạn ngữ cảnh của nhiều mô hình Transformer gốc (512–1024 tokens). Điều này dẫn đến mất ngữ cảnh toàn cục, làm giảm chất lượng tóm tắt.

  Ví dụ: Một bài báo về thử nghiệm lâm sàng giai đoạn III có thể có phần phương pháp, kết quả và thảo luận cách xa nhau; nếu mô hình chỉ xử lý 1024 tokens, nó có thể bỏ lỡ kết luận quan trọng về tác dụng phụ [1].
- Thuật ngữ chuyên môn và entity y khoa (Domain-specific Entities): Văn bản chứa entity phức tạp (tên thuốc, mã ICD-10, tên xét nghiệm, viết tắt) cần được giữ nguyên chính xác. Lỗi entity có thể dẫn đến hậu quả nghiêm trọng (ví dụ: nhầm "aspirin 81mg" thành "aspirin 325mg" trong tóm tắt hướng dẫn điều trị).

  Ví dụ: Trong tóm tắt về bệnh tiểu đường, mô hình có thể hallucinate "insulin glargine" thành "insulin aspart" nếu không được huấn luyện domain adaptation [14], [16].
- Hallucination và faithfulness (Faithfulness Issues): Mô hình abstractive thường tạo

thông tin không có trong nguồn (intrinsic: mâu thuẫn nội tại; extrinsic: thêm thông tin ngoài nguồn). Trong y tế, hallucination có thể gây hại (ví dụ: tóm tắt sai liều lượng thuốc hoặc hiệu quả điều trị).
Ví dụ: Mô hình có thể tóm tắt "Thuốc X giảm nguy cơ tử vong 20%" thành "Thuốc X giảm nguy cơ tử vong 50%" dù nguồn chỉ 20% – lỗi này phổ biến ở LLM zero-shot [17].

- Đa ngôn ngữ và domain shift: Tài liệu y tế chủ yếu tiếng Anh, nhưng ứng dụng thực tế (như tại Việt Nam) cần xử lý hỗn hợp hoặc dịch sang tiếng Việt, dẫn đến domain shift và mất thông tin.
  Ví dụ: Một bài báo PubMed tiếng Anh về "hypertension management" khi tóm tắt sang tiếng Việt có thể nhầm "blood pressure control" với các thuật ngữ địa phương [12].
- Đánh giá chất lượng: Metrics tự động như ROUGE (đo overlap n-gram) không phát hiện tốt hallucination hoặc entity error; cần kết hợp BERTScore, METEOR, human evaluation (faithfulness, entity precision) và metrics y tế chuyên biệt (ví dụ: MedFACT) [13], [15].

Các thách thức này đang được giải quyết dần nhờ mô hình long-document (như LongT5), PEFT (LoRA/QLoRA) và kỹ thuật mitigation hallucination (prompt engineering, retrieval-augmented generation) [1], [18].

1.4. Các mô hình Transformer và bài toán tóm tắt văn bản độ dài lớn

1.4.1. Từ T5 đến LongT5: Bước tiến trong xử lý ngữ cảnh quy mô lớn

Kiến trúc Transformer nguyên bản dựa trên cơ chế Full Self-Attention, nơi mỗi token đầu vào phải tương tác với tất cả các token còn lại. Điều này dẫn đến sự bùng nổ về chi phí tính toán và bộ nhớ theo hàm bậc hai $O(n^2)$ [19]. Theo khảo sát của Liu và các cộng sự (2024), đây chính là "điểm nghẽn" khiến các mô hình như T5 tiêu chuẩn thường bị giới hạn ở ngưỡng 512 hoặc 1024 tokens, không thể bao quát hết nội dung của các tài liệu dài như báo cáo tài chính hay nghiên cứu khoa học [21].Để giải quyết thách thức này, LongT5 được giới thiệu như một sự nâng cấp chiến lược, tích hợp cơ chế Transient Global (TGlobal) Attention [22]. Thay vì tính toán ma trận chú ý toàn bộ, LongT5 phân tách quá trình này thành hai thành phần:Local Attention: Tập trung vào các token trong phạm vi lân cận để duy trì tính liên kết cục bộ và ngữ pháp.Global Attention: Sử dụng các "global tokens" được tổng hợp thông qua phép lấy trung bình các khối (block-based summation). Các token này đóng vai trò là "trạm trung chuyển" thông tin, giúp các vị trí cách xa nhau trong văn bản vẫn có thể tương tác với chi phí

thấp [23].Nhờ cải tiến này, LongT5 có thể mở rộng cửa sổ ngữ cảnh lên tới 16.384 tokens mà vẫn duy trì hiệu suất tính toán tuyến tính, cho phép mô hình "đọc" toàn bộ một bài báo khoa học trong một lần xử lý duy nhất thay vì phải cắt nhỏ văn bản — một kỹ thuật vốn thường làm mất đi tính nhất quán của nội dung [24].

1.4.2. Hiệu năng của LongT5 trong miền tri thức Y sinh

Lĩnh vực y sinh đặt ra những thách thức đặc thù cho bài toán tóm tắt: từ vựng chuyên ngành dày đặc, cấu trúc bài viết chặt chẽ (IMRaD) và yêu cầu khắt khe về tính chính xác của thông tin y học. Các nghiên cứu từ năm 2023 đến đầu 2025 đã chứng minh LongT5 là ứng cử viên hàng đầu cho domain này thông qua các kết quả thực nghiệm quan trọng:

- **Sự vượt trội trên tập dữ liệu PubMed:** Các thực nghiệm trên biến thể longt5-tglobal-large-16384-pubmed cho thấy khi xử lý các tài liệu y khoa dài, mô hình không chỉ đạt điểm ROUGE cao hơn các mô hình truyền thống mà còn thể hiện khả năng "suy luận" giữa các phần khác nhau của bài báo (ví dụ: kết nối giả thuyết ở phần Introduction với kết quả ở phần Results) [26].
- **Khả năng thay thế mô hình đóng (LLMs):** Một nghiên cứu tiêu biểu vào năm 2023 đã tinh chỉnh LongT5 trên tập dữ liệu **MedReview** (gồm các bài tổng quan hệ thống). Kết quả cho thấy với kích thước tham số nhỏ hơn nhiều lần, LongT5 sau khi được fine-tune đạt hiệu suất tóm tắt tương đương với GPT-3.5 trong khi vẫn đảm bảo tính bảo mật dữ liệu và chi phí vận hành thấp — một yếu tố then chốt trong ứng dụng y tế thực tế [25], [27].
- **Tính ổn định với dữ liệu nhiễu:** Các khảo sát mới nhất năm 2025 chỉ ra rằng cơ chế chú ý toàn cục của LongT5 giúp mô hình ít bị ảnh hưởng bởi các đoạn văn bản "nhiễu" hoặc thông tin phụ trợ thường thấy trong các bài báo PubMed, từ đó trích xuất được các thực thể y học (thuốc, triệu chứng, chỉ số) một cách chính xác hơn trong bản tóm tắt [28].

1.4.3. Áp dụng thực tế trong đồ án

Trong phạm vi đồ án này, mô hình *long-t5-tglobal-base-booksum* được chọn làm nền tảng khởi đầu. Mặc dù được huấn luyện trên dữ liệu văn học (BookSum), nhưng khả năng nắm bắt dòng thời gian và mối quan hệ nhân quả trong các câu chuyện dài của mô hình này lại tương đồng một cách kỳ lạ với cách triển khai lập luận trong các bài báo khoa học dài [29].

Việc áp dụng phương pháp PEFT (Parameter-Efficient Fine-Tuning) lên mô hình này cho phép chúng ta:

- Kế thừa khả năng tóm tắt trừu tượng (abstractive) thượng thừa từ tiền huấn luyện.

- Chuyên biệt hóa mô hình vào miền y tế của PubMed mà không cần cập nhật toàn bộ hàng tỷ tham số, giúp tiết kiệm tài nguyên GPU trong khi vẫn đạt được độ chính xác mong muốn.

1.5. Parameter-Efficient Fine-Tuning (PEFT) trong domain y tế

1.5.1. Khái niệm PEFT

Trong kỷ nguyên của các mô hình ngôn ngữ lớn (LLMs), việc tinh chỉnh toàn bộ tham số (Full Fine-tuning) trở nên cực kỳ tốn kém về tài nguyên tính toán và lưu trữ. Parameter-Efficient Fine-Tuning (PEFT) nổi lên như một giải pháp chiến lược, cho phép thích nghi các mô hình tiền huấn luyện khổng lồ với các tác vụ chuyên biệt chỉ bằng cách cập nhật một lượng rất nhỏ tham số bổ sung [30].

Kỹ thuật tiêu biểu nhất là **LoRA (Low-Rank Adaptation)**, hoạt động bằng cách chèn các ma trận phân rã hạng thấp (low-rank matrices) vào các lớp chú ý của Transformer. Một bước tiến xa hơn là QLoRA, kết hợp định lượng 4-bit (4-bit quantization) để giảm thiểu tối đa yêu cầu về bộ nhớ VRAM mà vẫn duy trì độ chính xác gần như tương đương với full fine-tuning [31]. Ưu điểm cốt lõi của PEFT không chỉ nằm ở việc tiết kiệm GPU, mà còn giúp ngăn chặn hiện tượng Catastrophic Forgetting (quên kiến thức cũ), giúp mô hình giữ lại các kỹ năng ngôn ngữ tổng quát từ quá trình pre-train trong khi vẫn học được các tri thức chuyên sâu của domain mới [30].

1.5.2. Hiệu quả của PEFT trong tóm tắt văn bản y khoa

Việc áp dụng PEFT vào miền y tế (Medical Domain) đã ghi nhận nhiều kết quả đột phá trong giai đoạn 2023-2025, đặc biệt là với các tập dữ liệu có quy mô giới hạn hoặc đòi hỏi tính chuyên môn cao:

- **Vượt trội so với chuyên gia:** Các nghiên cứu năm 2024 chỉ ra rằng các mô hình ngôn ngữ được "thích nghi" bằng LoRA trong tác vụ tóm tắt bệnh án lâm sàng (Clinical Summarization) không chỉ vượt qua các mô hình truyền thống mà còn đạt điểm số đánh giá từ chuyên gia y tế cao hơn so với các bản tóm tắt thủ công, nhờ khả năng lọc nhiễu và trích xuất thực thể chính xác [32].
- **Đa tác vụ y tế:** Sự xuất hiện của các kiến trúc như MOELoRA (kết hợp Mixture-of-Experts với LoRA) đã cho thấy khả năng xử lý đồng thời nhiều nhiệm vụ y tế khác nhau (như tóm tắt, trả lời câu hỏi và phân loại mã bệnh) trên cùng một mô hình nền tảng, giúp tối ưu hóa luồng công việc trong bệnh viện [33].
- **Thích nghi với dữ liệu ít:** Các khảo sát mới nhất năm 2025 nhấn mạnh rằng PEFT cực kỳ hiệu quả trên các tập dữ liệu y tế nhỏ hoặc đặc thù, nơi mà việc

full fine-tune thường dẫn đến hiện tượng quá khớp (overfitting) và làm hỏng cấu trúc ngữ nghĩa của mô hình [34].

1.5.3. Cơ chế kỹ thuật của LoRA và giải pháp tối ưu hóa tài nguyên cho LongT5

Trong phạm vi đồ án, kỹ thuật **LoRA (Low-Rank Adaptation)** được lựa chọn làm phương pháp cốt lõi để thích nghi mô hình LongT5. Thay vì thay đổi toàn bộ trọng số của mô hình tiền huấn luyện, LoRA giả định rằng quá trình học thích nghi miền (domain adaptation) có "hạng nội tại" (intrinsic dimension) thấp [30].

Cơ chế hoạt động: Đối với một lớp ma trận trọng số đóng băng $W0 \in R^{d\times k}$, LoRA thực hiện cập nhật thông qua việc chèn thêm hai ma trận hạng thấp $W0 + \Delta W = W0 + BA$, trong đó $B \in R^{d\times r}$, $A \in R^{r\times k}$ (với $r << \min(d, k)$). Quá trình lan truyền tiến (forward pass) được sửa đổi thành:

$$h = W0x + \Delta W x = W0x + BAx$$

Trong đó, ma trận A được khởi tạo theo phân phối Gaussian và B khởi tạo bằng 0 để đảm bảo tại thời điểm bắt đầu huấn luyện $\Delta W= 0$ và mô hình vẫn giữ nguyên hiệu suất của bản pre-trained [30].

Tối ưu hóa chi phí tính toán trong đồ án:

Việc áp dụng LoRA thay vì tinh chỉnh toàn phần (Full Fine-tuning) mang lại những lợi ích cụ thể về tài nguyên:

- **Giảm thiểu tham số huấn luyện**: Với mô hình long-t5-tglobal-base, thay vì phải cập nhật khoảng 247 triệu tham số, việc áp dụng LoRA (thường với hạng r=8 hoặc 16) chỉ yêu cầu huấn luyện khoảng 0.5% - 1% tổng số tham số. Điều này giúp giảm đáng kể dấu chân bộ nhớ (memory footprint) của trình tối ưu hóa (optimizer states) như AdamW [31].
- **Tiết kiệm bộ nhớ VRAM**: Khi xử lý văn bản PubMed với chiều dài lên tới 16.384 tokens, bộ nhớ dành cho các lớp kích hoạt (activations) của LongT5 là cực lớn. LoRA giúp cắt giảm lượng bộ nhớ cần thiết để lưu trữ gradient cho các lớp trọng số cố định, cho phép huấn luyện mô hình trên các GPU phổ thông như RTX 3090 (24GB VRAM) thay vì cần đến các hệ thống máy chủ A100/H100 đắt đỏ [34].
- **Loại bỏ độ trễ khi suy luận**: Điểm đặc biệt của LoRA là các ma trận A và B có thể được "gộp" (merge) trực tiếp vào W0 sau khi huấn luyện xong. Điều này có nghĩa là khi triển khai mô hình tóm tắt bài báo y khoa, chúng ta không phải

chịu thêm bất kỳ chi phí tính toán hay độ trễ nào so với mô hình gốc, điều mà các phương pháp như Adapter thường gặp phải [30].

- **Tính linh hoạt và lưu trữ**: Thay vì lưu nhiều bản sao mô hình nặng hàng GB cho mỗi tác vụ, chúng ta chỉ cần lưu trữ các "LoRA weights" cực nhẹ (thường chỉ vài chục MB). Điều này đặc biệt phù hợp với môi trường nghiên cứu tại Việt Nam, nơi việc tối ưu hóa hạ tầng lưu trữ và tính toán là ưu tiên hàng đầu [35].

1.6. Dataset và benchmark trong BTS

Dataset đóng vai trò quan trọng trong việc huấn luyện, đánh giá và so sánh các mô hình tóm tắt văn bản y tế (Biomedical Text Summarization - BTS). Các dataset trong lĩnh vực này thường bao gồm các bài báo khoa học, báo cáo lâm sàng hoặc hồ sơ bệnh án, với cặp input (văn bản đầy đủ) và output (tóm tắt tham chiếu, thường là abstract hoặc phần kết luận). Chúng giúp mô hình học cách xử lý đặc thù y tế như entity chuyên môn, thuật ngữ dày đặc và ngữ cảnh dài. Dưới đây là tổng quan về các dataset phổ biến nhất, tập trung vào những phù hợp với abstractive summarization và long-document.

1.6.1. PubMed Article Summarization dataset (ccdv/pubmed-summarization)

Đây là dataset phổ biến nhất và được sử dụng rộng rãi cho nhiệm vụ abstractive summarization trên bài báo y khoa từ PubMed. Dataset được xây dựng từ kho PubMed Open Access, bao gồm các bài báo khoa học y sinh đầy đủ (article) làm input và abstract (tóm tắt) làm ground truth summary [36].

- **Chi tiết**: Dataset có kích thước khoảng 100K - 1M mẫu (train/validation/test), với input dài (thường hàng nghìn tokens) và abstract ngắn gọn, tập trung vào ý chính như mục tiêu nghiên cứu, phương pháp, kết quả và kết luận. Dữ liệu được tiền xử lý (pre-tokenized), dễ tích hợp với thư viện Hugging Face Transformers (qua run_summarization.py script).
- **Ưu điểm**: Phù hợp cho long-document summarization, phản ánh thực tế tài liệu y tế (bài báo PubMed thường dài và phức tạp). Dataset này được sử dụng trong nhiều nghiên cứu về mô hình Transformer-based, giúp đánh giá khả năng tổng hợp thông tin từ văn bản dài mà không mất ngữ cảnh.
- **Ví dụ**: Một input có thể là toàn bộ phần Introduction + Methods + Results + Discussion của bài báo về thử nghiệm lâm sàng, và summary là abstract khoảng 200-300 từ.
- **Ứng dụng**: Dataset lý tưởng cho fine-tuning các mô hình như LongT5, vì hỗ

trợ input dài và domain y tế chuyên sâu [36].

1.6.2. Benchmark và metrics đánh giá

Đánh giá BTS thường kết hợp metrics tự động (automatic) và đánh giá con người (human evaluation), vì metrics lexical có thể không phát hiện hallucination hoặc entity error.

- Metrics tự động phổ biến:
  - ROUGE (Recall-Oriented Understudy for Gisting Evaluation): Đo overlap n-gram (ROUGE-1/2/L), phổ biến nhất cho summarization. ROUGE-L tập trung vào longest common subsequence, phù hợp tóm tắt dài [1], [13].
  - BERTScore: Đo semantic similarity bằng cosine similarity vector từ BERT/RoBERTa, tốt hơn ROUGE ở việc xử lý paraphrase và synonym (ví dụ: "hypertension" và "high blood pressure"). Thường dùng F1-score [37].
  - METEOR: Kết hợp precision/recall với stemming, synonym matching và penalty cho order, tốt cho tóm tắt y tế nơi từ đồng nghĩa phổ biến [37].
  - Các metrics khác: BLEU (cho machine translation-like), chrF (character n-gram), hoặc entity-level metrics (entity precision/recall để đo faithfulness entity y khoa).
- Human evaluation: Đánh giá faithfulness (trung thực nguồn), entity accuracy (đúng entity như tên thuốc, mã bệnh), completeness (đủ ý), coherence (mạch lạc), và hallucination (không tạo thông tin sai). Thường dùng Likert scale hoặc pairwise comparison. Trong y tế, human eval rất quan trọng vì metrics tự động như ROUGE/BERTScore có correlation thấp với faithfulness [17].

1.6.3. Áp dụng vào đồ án

Trong đồ án "Xây dựng ứng dụng tóm tắt văn bản y tế", dataset chính được sử dụng là PubMed Article Summarization (ccdv/pubmed-summarization) để fine-tune base model long-t5-tglobal-base-booksum bằng PEFT. Dataset này phù hợp vì input dài (PubMed articles), domain y tế tiếng Anh, và abstract làm ground truth abstractive.

Quá trình đánh giá sẽ sử dụng ROUGE (1/2/L). Human evaluation với biomedical ner (Named Entity Recognition) sẽ tập trung faithfulness và entity accuracy để đảm bảo tóm tắt an toàn cho ứng dụng y tế. Việc fine-tune trên PubMed giúp mô hình

thích nghi tốt hơn với entity y khoa, giảm hallucination, và hỗ trợ người dùng Việt Nam (bác sĩ/nhà nghiên cứu) tóm tắt nhanh tài liệu PubMed tiếng Anh [36].

1.7. Tổng quan về ngôn ngữ lập trình Kotlin và SwiftUI:

Trong bối cảnh phát triển ứng dụng di động hiện đại, đặc biệt là các ứng dụng y tế xử lý dữ liệu nhạy cảm như văn bản y khoa (báo cáo bệnh án, bài báo PubMed, tóm tắt abstractive), việc lựa chọn ngôn ngữ và framework giao diện cần đảm bảo tính an toàn, hiệu suất, dễ bảo trì và trải nghiệm người dùng mượt mà. Đồ án này sử dụng Kotlin cho Android (kết hợp XML) và SwiftUI cho iOS để xây dựng phiên bản đa nền tảng, tận dụng ưu điểm của từng hệ sinh thái trong việc hiển thị văn bản dài, highlight entity y khoa (tên thuốc, triệu chứng, mã ICD-10), và quản lý trạng thái động (đang tải tóm tắt từ backend, lỗi xử lý, kết quả hiển thị) [38], [42].

1.7.1. Ngôn ngữ Kotlin trong phát triển Android

- Trong dự án Xây dựng ứng dụng tóm tắt văn bản y tế, nền tảng Android được phát triển bằng ngôn ngữ lập trình Kotlin kết hợp với XML để xây dựng giao diện người dùng. Đây là cách tiếp cận truyền thống nhưng ổn định và hiệu quả, đặc biệt phù hợp với các ứng dụng xử lý nội dung chuyên sâu, yêu cầu giao diện rõ ràng, dễ đọc và dễ kiểm soát luồng hiển thị thông tin y khoa.

- Kotlin mang lại nhiều lợi ích quan trọng như cú pháp ngắn gọn, an toàn null (null safety) và khả năng tương thích cao với Java, giúp tận dụng tối đa hệ sinh thái Android hiện có. Điều này đặc biệt hữu ích trong việc triển khai các chức năng xử lý dữ liệu văn bản, gọi API tóm tắt, quản lý trạng thái tải dữ liệu và xử lý lỗi. Trong khi đó, việc xây dựng giao diện bằng XML giúp mô phỏng chính xác thiết kế từ Figma sang ứng dụng thực tế, hỗ trợ tốt cho các màn hình có bố cục phức tạp như hiển thị văn bản y tế dài, kết quả tóm tắt, so sánh nội dung gốc và nội dung rút gọn (hướng phát triển), hoặc đánh dấu các thông tin y khoa quan trọng (hướng phát triển).

- Ưu điểm nổi bật khi dùng Kotlin + XML trong ứng dụng tóm tắt văn bản y tế:

- **An toàn và giảm lỗi runtime**: Null safety loại bỏ NullPointerException – rủi ro cao khi xử lý dữ liệu y tế từ người dùng (ví dụ: văn bản nhập tay hoặc từ PDF), đảm bảo entity quan trọng (tên thuốc, triệu chứng) không bị null dẫn đến crash hoặc hiển thị sai [38]. Ví dụ: Khi hiển thị tóm tắt, Kotlin giúp kiểm tra null an toàn trước khi bind dữ liệu vào TextView.
- **Tái sử dụng layout và kiểm soát giao diện**: XML cho phép định nghĩa các component tái sử dụng như CardView cho khung văn bản gốc/tóm tắt, RecyclerView cho lịch sử tóm tắt, và ConstraintLayout cho bố cục phức tạp

(so sánh nội dung gốc, tóm tắt, highlight entity bằng màu sắc – Hướng phát triển). Điều này đảm bảo nội dung y tế được trình bày rõ ràng, dễ đọc (font lớn, khoảng cách dòng rộng, dark mode hỗ trợ - Hướng phát triển), giảm nguy cơ hiểu sai thông tin lâm sàng [39].

- **Tách biệt logic và UI theo MVVM**: Kotlin xử lý ViewModel (quản lý trạng thái: loading, success, error khi tóm tắt), LiveData/Flow cho cập nhật realtime, trong khi XML định nghĩa UI – phù hợp với ứng dụng y tế cần dễ test (unit test logic tóm tắt) và bảo trì (thêm tính năng highlight entity mới) [40].
- **Hỗ trợ coroutines và Flow:** Xử lý bất đồng bộ mượt mà khi gọi mô hình AI (LongT5 inference), tải văn bản dài hoặc đồng bộ lịch sử tóm tắt với Room database, tránh blocking UI thread – rất quan trọng cho trải nghiệm người dùng khi xử lý tài liệu y tế [41]

1.7.2. SwiftUI và giao diện người dung cho hệ sinh thái iOS

- SwiftUI là framework do Apple phát triển nhằm xây dựng giao diện người dùng theo hướng hiện đại, trực quan và có tính khai báo (declarative). Trong dự án xây dựng ứng dụng tóm tắt văn bản y tế, SwiftUI được sử dụng để phát triển phiên bản iOS của ứng dụng, với mục tiêu mang lại trải nghiệm mượt mà, nhất quán và tối ưu cho việc đọc, phân tích và tương tác với nội dung y khoa.

- So với UIKit truyền thống, SwiftUI cho phép mô tả giao diện bằng cú pháp Swift ngắn gọn, dễ hiểu và gắn chặt với dữ liệu động thông qua các cơ chế như @State, @Binding và ObservableObject. Cách tiếp cận này đặc biệt phù hợp với ứng dụng xử lý văn bản y tế, nơi giao diện cần phản ánh liên tục các trạng thái như: đang xử lý tóm tắt, hiển thị kết quả, cập nhật nội dung đầu vào hoặc làm nổi bật các thông tin quan trọng. Ngoài ra, SwiftUI hỗ trợ tốt việc chuyển đổi thiết kế từ Figma sang giao diện thực tế, đảm bảo tính nhất quán về bố cục và trải nghiệm người dùng giữa các nền tảng.

- Ưu điểm khi sử dụng SwiftUI trong dự án:

- **Cập nhật giao diện thời gian thực và declarative**: Mô tả UI theo trạng thái (state-driven), tự động redraw khi dữ liệu thay đổi (ví dụ: @State var summaryText cập nhật Text view ngay khi backend trả tóm tắt), giảm boilerplate code so với UIKit – phù hợp cho hiển thị kết quả abstractive động [42]. Ví dụ: Khi người dùng nhập văn bản bệnh án, SwiftUI tự cập nhật preview tóm tắt mà không cần reload view.
- **Preview trực quan và thiết kế nhanh**: Xcode Preview cho phép xem thay đổi realtime (từ Figma sang code), hỗ trợ dark mode, dynamic type và accessibility (VoiceOver đọc entity y khoa rõ ràng) – đảm bảo giao diện y tế

dễ tiếp cận cho bác sĩ và bệnh nhân [43].

- **Tích hợp sâu với SwiftUI ecosystem**: Hỗ trợ Combine/AsyncSequence cho bất đồng bộ (gọi API, hiển thị progress bar), và @Environment cho inject dữ liệu toàn cục (như user preferences về ngôn ngữ tóm tắt) [44]. SwiftUI còn hỗ trợ đa nền tảng (iPhone/iPad/macOS), mở rộng ứng dụng sang Mac cho nhà nghiên cứu xem tóm tắt PubMed trên màn hình lớn.
- **Hiệu suất và accessibility built-in**: SwiftUI tối ưu cho modern devices (iPhone 12+), với built-in support cho large text, high contrast – quan trọng trong y tế để tránh hiểu sai thông tin (ví dụ: triệu chứng được highlight rõ) [45].

1.8. FastAPI và kiến trúc RESTful API

1.8.1. Tổng quan về FastAPI

FastAPI là framework web Python hiện đại (ra mắt 2018), dựa trên Starlette (ASGI async) và Pydantic (type validation), nổi bật với hiệu năng cao (tương đương Node.js/Go), tự động OpenAPI docs (Swagger/ReDoc) và hỗ trợ async/await [46], [47]. Trong đồ án, FastAPI làm backend để xử lý request tóm tắt văn bản y tế (nhận input từ app di động, chạy LongT5 inference, trả tóm tắt JSON), tận dụng async cho concurrency cao khi nhiều người dùng gửi tài liệu dài [48].

Ưu điểm nổi bật trong ứng dụng AI/ML y tế:

- **Hiệu năng và async**: Xử lý nhiều request đồng thời (inference LongT5 trên GPU/CPU), phù hợp cho real-time summarization [46].
- **Type safety và auto-validation**: Pydantic validate input (văn bản y tế không rỗng, độ dài hợp lý), giảm lỗi runtime – quan trọng để tránh hallucination do input xấu [47].
- **Tích hợp Hugging Face**: Dễ load model LongT5/PEFT, deploy inference endpoint (/summarize) [48].
- **Auto docs và dễ test**: Swagger UI giúp test endpoint nhanh, hỗ trợ tích hợp frontend (Android/iOS) [46].

Nhược điểm: Async code phức tạp hơn cho người mới, CPU-heavy tasks (inference lớn) cần worker (Celery) – khắc phục bằng Uvicorn + GPU [47].

1.8.2. Kiến trúc RESTful API với FastAPI

RESTful API dựa trên resources và HTTP methods (GET/POST/PUT/DELETE tương ứng CRUD), stateless, sử dụng status code chuẩn (200 OK, 201 Created, 400 Bad Request, 401 Unauthorized) [49]. FastAPI triển khai RESTful tự nhiên: router

định nghĩa endpoint, dependency injection cho auth (JWT), Pydantic models cho request/response [46].

Áp dụng:

- Endpoint POST /summarize nhận

  {"text": "Nội dung bệnh án hoặc tài liệu y tế...",

  "mode": "short"}

- Kết quả trả về {"summary": "Nội dung tóm tắt..."}với status code và error handling cho input y tế không hợp lệ [50].

1.8.3. Mô hình MVC

- MVC là viết tắt của cụm từ "Model-View-Controller" [51]. Là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. MVC là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính.

- Mô hình MVC thường được chia làm 3 phần. Mỗi phần đảm bảo một vai trò và nhiệm vụ riêng biệt khác nhau.
- Model: Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng và là cầu nối giữa 2 thành phần bên dưới là View và Controller.
- View: Đây là phần giao diện dành cho người dùng.MVC là phương tiện hiển thị các đối tượng trong một ứng dụng.
- Controller: Là bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua View.

- Luồng xử lý trong MVC

- Khi một yêu cầu của khách hàng từ máy khách (Client) gửi đến Server. Thì bị Controller trong MVC chặn lại để xem đó là URL request hay sự kiện.
- Sau đó, Controller xử lý input của user rồi giao tiếp với Model trong MVC.
- Model chuẩn bị data và gửi lại cho Controller.
- Cuối cùng, khi xử lý xong yêu cầu thì Controller giữ dữ liệu trở lại View và hiển thị cho người dùng trên trình duyệt.

1.9. Tổng quan về cơ sở dữ liệu cục bộ Room, CoreData

1.5.1. Room Database (Android)

Room là thư viện lưu trữ dữ liệu cục bộ do Google phát triển, thuộc bộ Android Jetpack, hoạt động như một lớp trừu tượng (abstraction layer) trên SQLite. Room giúp đơn giản hóa việc thao tác với cơ sở dữ liệu bằng cách cung cấp các thành phần rõ ràng và an toàn kiểu dữ liệu (*type-safe*) [52].

Các thành phần chính của Room gồm:

- Entity: Biểu diễn bảng dữ liệu trong cơ sở dữ liệu.
- DAO (Data Access Object): Định nghĩa các phương thức truy vấn dữ liệu (insert, update, delete, query).
- Database: Lớp quản lý cơ sở dữ liệu và kết nối các entity với DAO.

Ưu điểm của Room:

- Hỗ trợ kiểm tra câu lệnh SQL tại thời điểm biên dịch.
- Tích hợp tốt với kiến trúc MVVM và LiveData / Flow.
- Giảm thiểu lỗi khi thao tác trực tiếp với SQLite.
- Phù hợp cho các ứng dụng Android cần lưu trữ lịch sử xử lý dữ liệu, chẳng hạn như kết quả tóm tắt văn bản y tế.

1.5.2. Core Data (iOS)

Core Data là framework quản lý dữ liệu cục bộ do Apple cung cấp, được sử dụng phổ biến trong các ứng dụng iOS. Core Data không chỉ đơn thuần là một cơ sở dữ liệu, mà là một hệ thống quản lý vòng đời đối tượng (object graph management), cho phép ánh xạ dữ liệu từ đối tượng Swift sang dạng lưu trữ bền vững, thường là SQLite [53].
Các thành phần chính của Core Data bao gồm:

- Managed Object Model: Định nghĩa cấu trúc dữ liệu và các thực thể.
- Managed Object Context: Quản lý trạng thái và vòng đời của các đối tượng.
- Persistent Store: Lớp lưu trữ dữ liệu vật lý (thường là SQLite).

Ưu điểm của Core Data:

- Tối ưu cho hệ sinh thái iOS và SwiftUI.
- Hỗ trợ quản lý dữ liệu phức tạp và quan hệ giữa các thực thể.
- Cơ chế cache và lazy loading giúp cải thiện hiệu năng.
- Phù hợp cho các ứng dụng cần lưu trữ và truy xuất dữ liệu cục bộ một cách hiệu quả.

1.10. Kết chương 1

Chương 1 đã trình bày các cơ sở lý thuyết nền tảng phục vụ cho việc xây dựng ứng dụng tóm tắt văn bản y tế dựa trên trí tuệ nhân tạo. Nội dung chương tập trung giới thiệu mô hình ngôn ngữ lớn (LLM), kiến trúc Transformer và vai trò của chúng trong bài toán tóm tắt văn bản. Đồng thời, chương cũng phân tích mô hình nền được lựa chọn là pszemraj/long-t5-tglobal-base-16384-book-summary, phù hợp với việc xử lý các tài liệu y tế có độ dài lớn. Kỹ thuật tinh chỉnh mô hình hiệu quả tham số (PEFT) với LoRA được trình bày nhằm giảm chi phí huấn luyện trong khi vẫn đảm bảo hiệu năng mô hình. Bên cạnh đó, các vấn đề liên quan đến tối ưu hóa, độ phức tạp tính toán và phương pháp

đánh giá mô hình bằng chỉ số ROUGE cũng được đề cập. Những nội dung lý thuyết này tạo nền tảng quan trọng cho việc thiết kế, triển khai và đánh giá hệ thống trong các chương tiếp theo của đồ án.

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Phân tích nghiệp vụ chính của người dùng

2.1.1. Nhập và quản lý tài liệu y tế

- Nhập văn bản y tế trực tiếp dưới dạng đoạn text dài.
- Tải lên tài liệu định dạng PDF hoặc DOC/DOCX.
- Trích xuất nội dung văn bản từ tài liệu tải lên để phục vụ cho các bước xử lý NLP tiếp theo.
- Hiển thị nội dung văn bản gốc để người dùng đối chiếu với kết quả phân tích.

2.1.2. Tóm tắt văn bản y tế

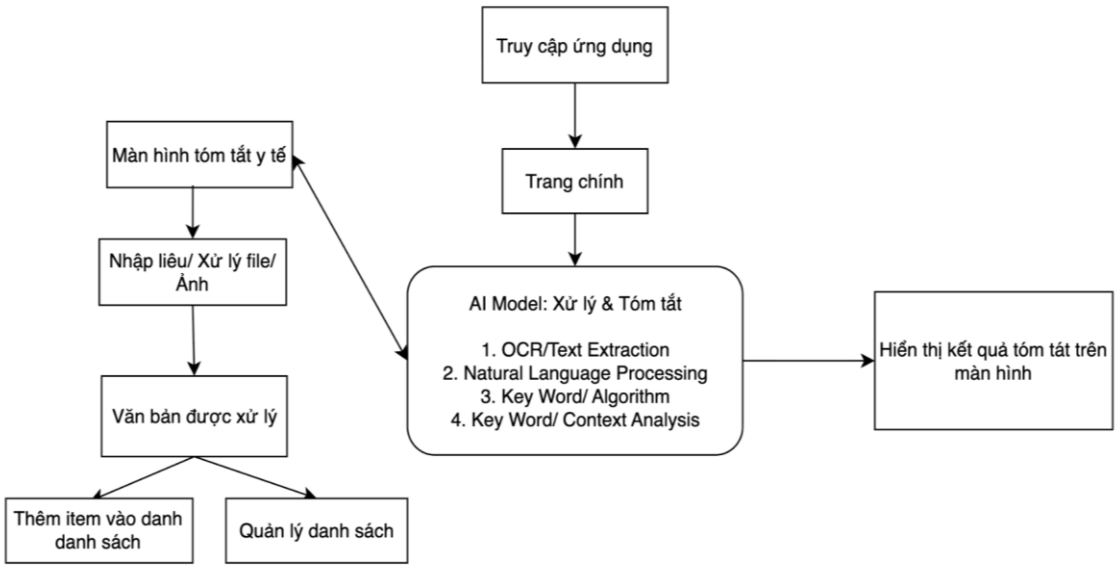
- Sinh bản tóm tắt ngắn gọn từ văn bản đầu vào bằng mô hình NLP.
- Hỗ trợ tóm tắt các loại tài liệu y tế như bài báo nghiên cứu, báo cáo lâm sàng, tài liệu học tập.
- Cho phép người dùng điều chỉnh mức độ tóm tắt (ngắn/trung bình).
- Hiển thị kết quả tóm tắt song song với văn bản gốc.

2.1.3. Tương tác chatbot

- Gửi câu hỏi tự nhiên liên quan đến tài liệu (ví dụ: "Các triệu chứng chính được đề cập trong tài liệu này là gì?").
- Nhận phản hồi được ngữ cảnh hóa dựa trên nội dung văn bản đã nhập.
- Hỗ trợ hội thoại nhiều lượt, giữ được ngữ cảnh trong phiên làm việc.
- Sinh câu hỏi ôn tập (quiz) dựa trên nội dung tài liệu y tế.

2.2. Thiết kế hệ thống

2.2.1. Sơ đồ nguyên lý hoạt động

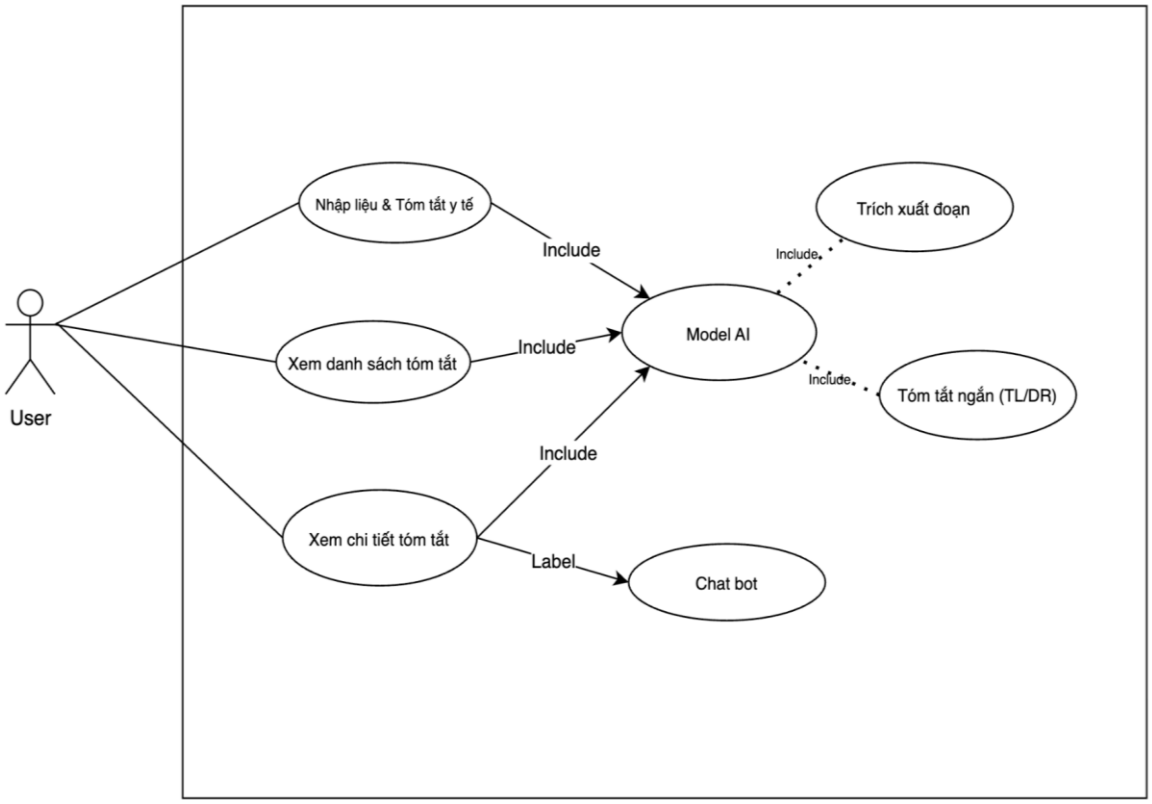


Hình 2.1. Sơ đồ nguyên lý hoạt độngSơ đồ ca sử dụng (Usecase)

2.2.2. Sơ đồ ca sử dụng

- Sơ đồ ca sử dụng là một kỹ thuật được dùng trong kỹ thuật phần mềm và hệ thống để nắm bắt yêu cầu chức năng của hệ thống.

- Usecase tổng quát:



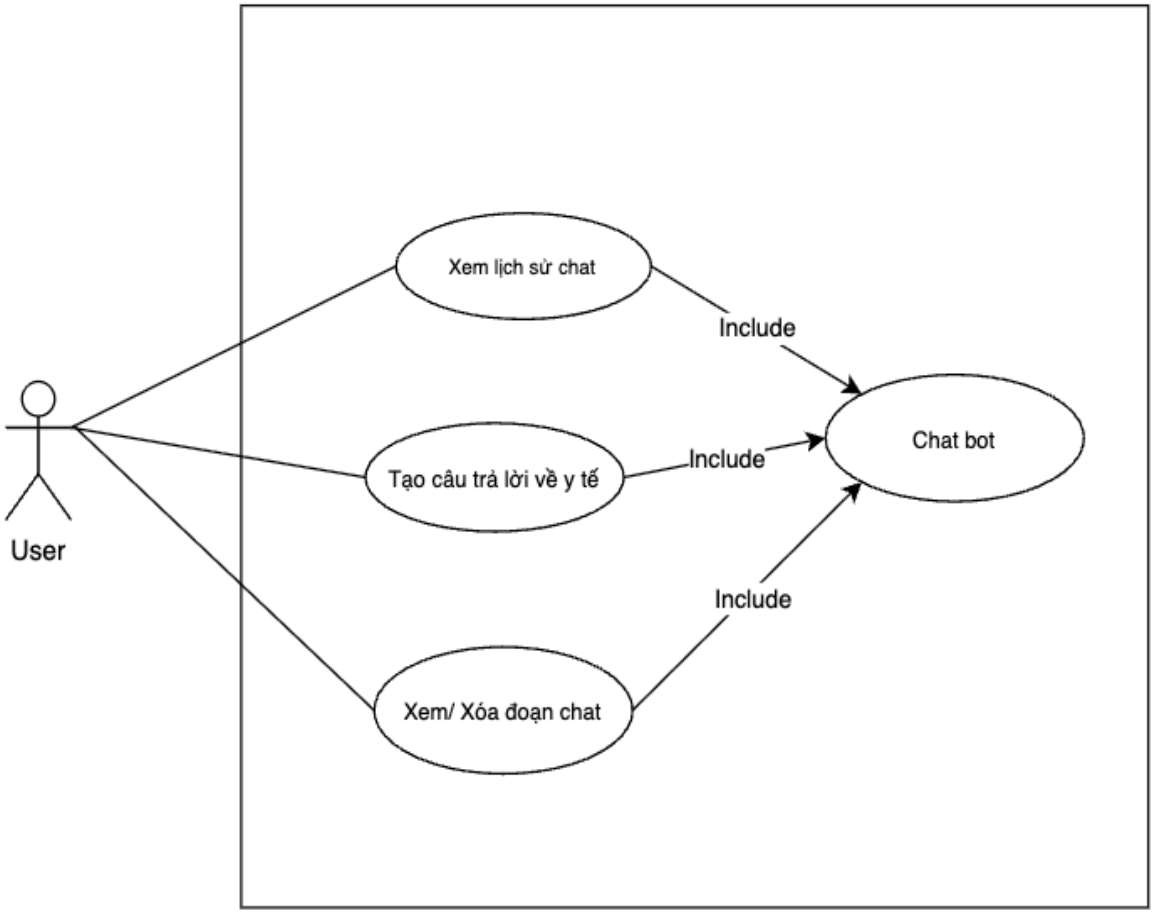
Hình 2.2. Sơ đồ Usecase tổng quát

Bảng 2.1. Đặc tả Usecase tổng quát

Mã Use Case	UC-01
Tên Use Case	Tổng quát hệ thống
Tác nhân	Người dùng
Mô tả	Cho phép người dùng sử dụng toàn bộ các chức năng chính của ứng dụng tóm tắt văn bản y tế, bao gồm nhập liệu văn bản, tóm tắt nội dung y khoa, trích xuất thông tin quan trọng và tương tác với chatbot AI hỗ trợ giải thích và tư vấn nội dung y tế
Điều kiện tiên quyết	Người dùng đã cài đặt ứng dụng trên thiết bị Android hoặc iOS và có kết nối Internet..

Luồng chính	1. Người dùng truy cập ứng dụng. 2. Người dùng nhập hoặc dán văn bản y tế (bệnh án, bài báo y khoa, hướng dẫn điều trị, kết quả xét nghiệm…). 3. Sử dụng các chức năng chính (UC02 → UC05)
Kết quả mong muốn	Người dùng nhanh chóng nắm được ý chính của văn bản y tế dài và phức tạp.

- User Trò chuyện Chatbot:

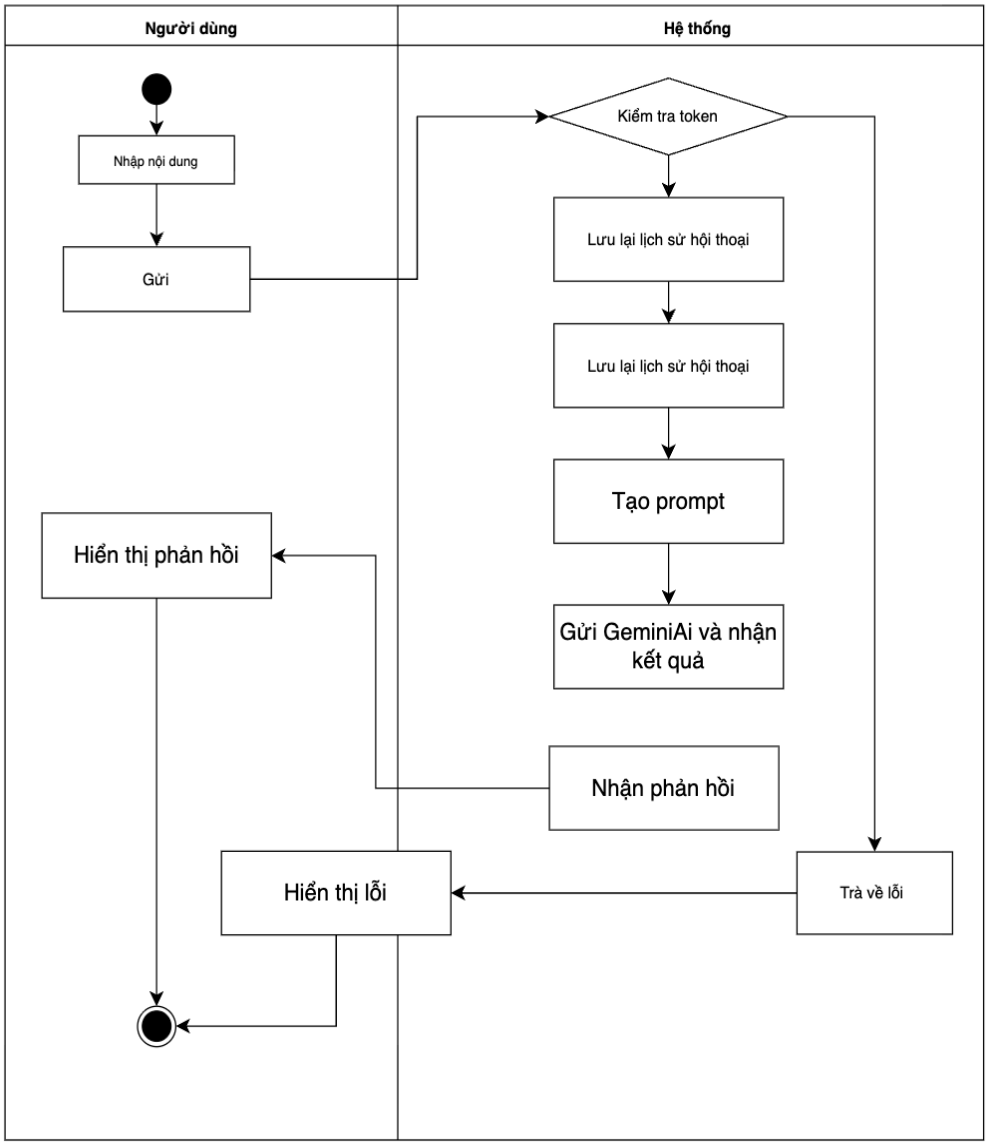


Hình 2.3. Sơ đồ Usecase Trò chuyện Chatbot

Bảng 2.2. Đặc tả Usecase Tương tác với Chatbot

Mã Use Case	UC-05
Tên Use Case	Tương tác với Chatbot
Tác nhân	Người dùng
Mô tả	Người dùng đặt câu hỏi tự nhiên về sức khỏe và y tế. Hệ thống gửi truy vấn đến API Gemini với prompt sức khỏe.. Dữ liệu sức khỏe và y tế được đính kèm để cá nhân hóa phản hồi. Chatbot có thể nhớ hội thoại và phản hồi liên tục nhiều lượt.
Điều kiện tiên quyết	Người dùng đã đăng nhập và có kết nối internet.
Luồng chính	1. Gửi câu hỏi như "Đâu là triệu chứng của cảm cúm"
	2. Backend xử lý, gọi Gemini API 3. Gemini trả lời phối đồ Hiển thị trả lời cho người dùng
Kết quả mong muốn	Người dùng nhận được lời khuyên để cải thiện sức khỏe và cho biết tình trạng 1 cách hợp lí

- Sơ đồ hoạt động chức năng trò chuyện với chatbot



Hình 2.4. Sơ đồ chức năng Tương tác với Chatbot

2.3. Kết chương 2

Chương 2 đã trình bày quá trình phân tích nghiệp vụ và thiết kế hệ thống cho ứng dụng tóm tắt văn bản y tế. Thông qua các sơ đồ ca sử dụng, các chức năng chính của hệ thống được xác định rõ ràng, đặc biệt là các chức năng liên quan đến việc nhập văn bản, tóm tắt nội dung và tương tác với chatbot tư vấn sức khỏe. Chương cũng mô tả luồng hoạt động tổng thể của hệ thống, thể hiện cách người dùng tương tác với ứng dụng và cách chatbot xử lý yêu cầu thông qua mô hình trí tuệ nhân tạo. Những nội dung phân tích và thiết kế trong chương này là cơ sở quan trọng cho việc triển khai, tích hợp chatbot và hoàn thiện ứng dụng ở các chương tiếp theo, đảm bảo hệ thống hoạt động ổn định và đáp ứng đúng mục tiêu đề ra.

CHƯƠNG 3: TRIỂN KHAI VÀ ĐÁNH GIÁ

3.1. Môi trường và công cụ lập trình

- Backend: FastAPI.
- Frontend Android: Sử dụng Kotlin kết hợp XML để xây dựng giao diện hiện đại, dễ tùy biến.
- Frontend iOS: Sử dụng SwiftUI giúp khai thác tối đa khả năng native UI trên iOS.
- Cơ sở dữ liệu: Room, CoreData.
- Công cụ hỗ trợ: Android Studio, Xcode, Cursor, Github.
- AI & Chatbot: Mô hình tóm tắt văn bản y tế và chatbot tư vấn sức khoẻ

3.2. Mô tả chức năng kết quả đã đạt được

3.2.1. Chuẩn bị dữ liệu

Dữ liệu sử dụng: PubMed Article Summarization Dataset

a. Giới thiệu về PubMed Article Summarization Dataset

- PubMed Article Summarization Dataset là tập dữ liệu phổ biến trong lĩnh vực xử lý ngôn ngữ tự nhiên, được xây dựng từ các bài báo khoa học y sinh học trên cơ sở dữ liệu **PubMed**. Mỗi mẫu dữ liệu bao gồm hai thành phần chính: **nội dung bài báo y tế (article)** và **bản tóm tắt tương ứng (abstract)** do các chuyên gia hoặc tác giả bài báo biên soạn. Các bài viết trong tập dữ liệu thường có độ dài lớn, chứa nhiều thuật ngữ chuyên ngành, phù hợp cho bài toán tóm tắt văn bản dài trong lĩnh vực y tế.

b. Lý do lựa chọn PubMed Dataset

- Phù hợp với lĩnh vực y tế: Nội dung dữ liệu đến từ các bài báo khoa học y sinh, sát với mục tiêu xây dựng ứng dụng tóm tắt văn bản y tế.
- Chất lượng cao: Phần tóm tắt (abstract) được viết bởi các nhà nghiên cứu, đảm bảo tính chính xác, khoa học và nhất quán.
- Văn bản dài: Độ dài của bài báo phù hợp với mô hình Long-T5, cho phép khai thác hiệu quả khả năng xử lý ngữ cảnh dài.
- Được sử dụng rộng rãi: PubMed là tập dữ liệu chuẩn trong nhiều nghiên cứu về tóm tắt văn bản, giúp kết quả của đồ án có tính tham chiếu và so sánh với các công trình trước đó.

c. Tiền xử lý dữ liệu

- Trích xuất nội dung bài báo (*article*) và bản tóm tắt (*abstract*) từ tập dữ liệu PubMed.
- Loại bỏ các mẫu dữ liệu bị thiếu thông tin hoặc có giá trị rỗng (*NaN*).

- Tính toán số lượng từ của mỗi bài báo và bản tóm tắt để phân tích phân bố độ dài văn bản.

- Sử dụng biểu đồ hộp (*boxplot*) để xác định các giá trị ngoại lai về độ dài.

- Loại bỏ các mẫu có độ dài vượt quá ngưỡng cho phép nhằm phù hợp với khả năng xử lý của mô hình.

- Giảm kích thước tập dữ liệu bằng cách lấy mẫu ngẫu nhiên và xáo trộn dữ liệu.

- Chia dữ liệu thành tập huấn luyện và tập kiểm tra.

- Token hóa văn bản đầu vào và đầu ra, giới hạn độ dài tối đa và chuẩn hóa dữ liệu trước khi đưa vào mô hình.

3.2.2. Chuẩn bị mô hình

- Tải mô hình nền và lượng tử hóa

```
bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_compute_dtype=torch.float16
)

model = AutoModelForSeq2SeqLM.from_pretrained(
    model_ckpt,
    quantization_config=bnb_config,
    device_map="auto"
)
```

Hình 3.1. Sơ đồ load base model + quantization 4-bit

- Áp dụng kỹ thuật PEFT với LoRA

```
peft_config = LoraConfig(
    r=16,
    lora_alpha=32,
    target_modules=["q", "v"],
    lora_dropout=0.1,
    bias="none",
    task_type="SEQ_2_SEQ_LM"
)

model = get_peft_model(model, peft_config)
```

Hình 3.2. Sơ đồ LoRA chèn vào Transformer attention

- Cấu hình tham số huấn luyện

```
training_args = Seq2SeqTrainingArguments(
    output_dir="./longt5_pubmed_peft",

    per_device_train_batch_size=1,
    per_device_eval_batch_size=1,
    gradient_accumulation_steps=4,

    learning_rate=5e-5,
    num_train_epochs=3,

    # === SAVE & LOG ===
    save_steps=500,
    save_total_limit=3,

    logging_steps=100,
    logging_dir="./logs",

    fp16=True,
    predict_with_generate=True,

    report_to="none"
)
```

Hình 3.3. Tham số huấn luyện mô hình

- Kết quả huấn luyện

Lr	Epochs	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum
5e-5	3	0.2678	0.2678	0.1522	0.2120
2e-4	3	0.2714	0.0567	0.1521	0.2146
1e-4	2	0.2690	0.0572	0.1522	0.2136

- Đánh giá:

Kết quả thực nghiệm cho thấy việc điều chỉnh tốc độ học trong khoảng từ 5e-5 đến 2e-4 không tạo ra sự khác biệt đáng kể về điểm số ROUGE. Mặc dù tốc độ học 2e-4 cho kết quả ROUGE nhỉnh hơn một chút, sự chênh lệch này là rất nhỏ (<0,5%) và không mang lại lợi ích rõ rệt khi tiếp tục giảm tốc độ học trong các giai đoạn huấn luyện sau. Điều này cho thấy PEFT adapter có xu hướng hội tụ sớm, và tốc độ học chủ yếu ảnh hưởng đến độ ổn định huấn luyện hơn là chất lượng tóm tắt cuối cùng. Do đó, tốc độ học 5e-5 được lựa chọn nhằm đảm bảo quá trình huấn luyện ổn định và bảo toàn tốt hơn kiến thức của mô hình cơ sở.

- Kiểm tra với NER y sinh (Biomedical Named Entity Recognition) với model "d4data/biomedical-ner-all":

```
 ARTICLE ENTITIES:
['##het', '##ilated room', '##ized', '##m', '##than', '##thasone', '##tus', '22 2c', 'an', 'ap
optosis', 'cells', 'cerebral', 'corticosteroids', 'dentate gyrus', 'dex', 'dexame', 'divide af
ter term', 'eu', 'g', 'high', 'hippocampus', 'human', 'hydrate', 'lung', 'neurons', 'normal',
'phosphate', 'pup', 'rat', 'steroids', 'vent', 'white', 'wi']

 GOLD ENTITIES:
['##amethasone', '##bino wistar', '##fepristone', '##op', '##term', '##tosis', '##us', 'al',
'ap', 'assist', 'dex', 'gcr', 'high', 'hip', 'hippo', 'hippocampi', 'hippocampus', 'increase
d', 'infants', 'inflammation', 'mi', 'neural progenitor cells', 'normal', 'of glucocorticoid r
eceptors', 'p', 'pre', 'pup', 'pups', 'rat', 'reduced', 'single', 'ventilation']

 GENERATED ENTITIES:
['##term', 'dex', 'dose', 'pre', 'preterm', 'pup', 'rat litter', 'single', 'timing']

✅ SUPPORTED ENTITIES (Gen ∩ Article):
['dex', 'pup']

❌ HALLUCINATED ENTITIES (Gen - Article):
['##term', 'dose', 'pre', 'preterm', 'rat litter', 'single', 'timing']

⚠️ MISSING ENTITIES (Gold - Gen):
['##amethasone', '##bino wistar', '##fepristone', '##op', '##tosis', '##us', 'al', 'ap', 'assi
st', 'gcr', 'high', 'hip', 'hippo', 'hippocampi', 'hippocampus', 'increased', 'infants', 'infl
ammation', 'mi', 'neural progenitor cells', 'normal', 'of glucocorticoid receptors', 'p', 'pup
s', 'rat', 'reduced', 'ventilation']
```

Hình 3.3. Màn hình hiển thị các entities

- Tính điểm hallucination:

```python
def hallucination_score_article(article_ents, gen_ents):
    if len(gen_ents) == 0:
        return 0.0
    return len(gen_ents - article_ents) / len(gen_ents)

score = hallucination_score_article(
    analysis["article_entities"],
    analysis["generated_entities"]
)

print(f"\n📊 HALLUCINATION SCORE (article-based): {score:.2f}")
```

Hình 3.4. Hàm tính điểm hallucination

- Kết quả hallucination score cho mẫu index 4096:
    - HALLUCINATION SCORE (article-based): 0.78

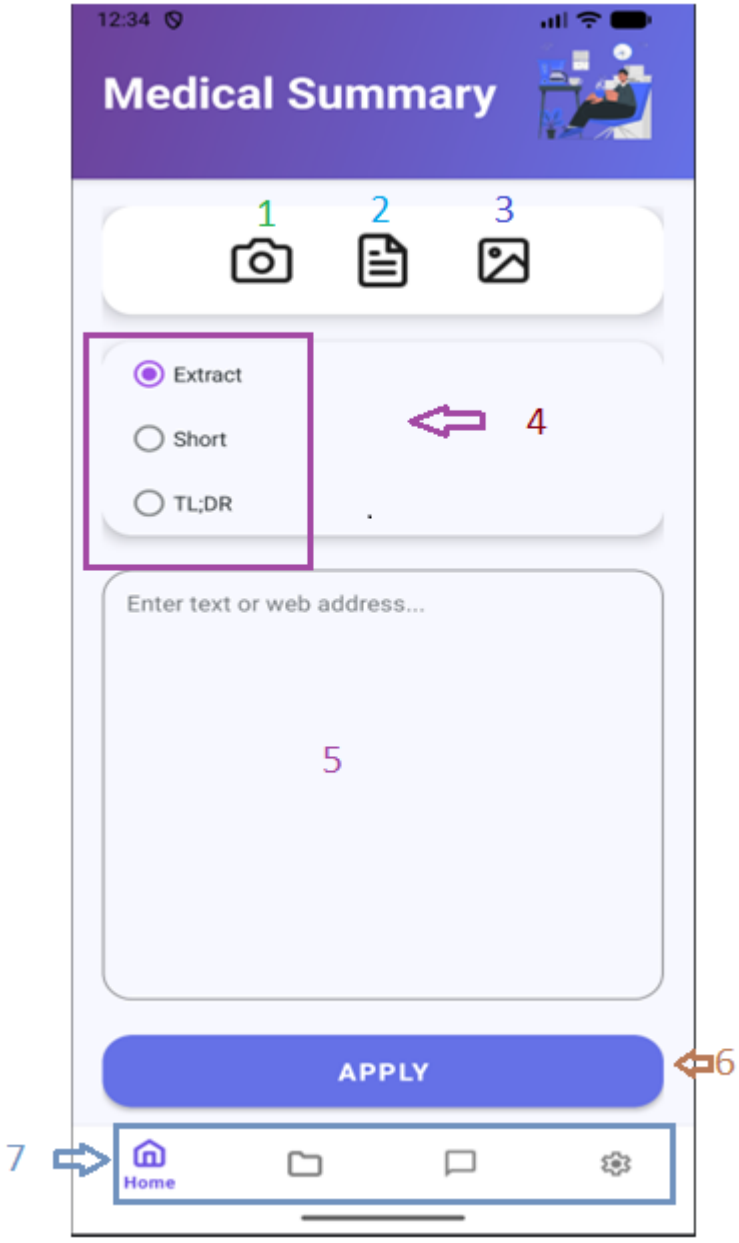
Kết luận: Mặc dù mô hình thể hiện khả năng tóm tắt và diễn đạt lại nội dung khá tốt, đảm bảo được ý nghĩa tổng quát của văn bản gốc, tuy nhiên việc thay đổi văn phong, mô hình có xu hướng viết lại hoặc biến đổi tên các thực thể y sinh, kèm theo lỗi chính tả hoặc khác biệt về cách biểu diễn so với bản tóm tắt tham chiếu

- > Làm giảm mức độ trùng khớp bề mặt giữa văn bản sinh và văn bản tham chiếu, từ đó làm tăng chỉ số hallucination, mặc dù nội dung cốt lõi trong nhiều trường hợp vẫn

giữ được tính hợp lý và đúng ngữ cảnh.

3.2.3. Giao diện chức năng của ứng dụng

- Màn hình chính



Hình 3.5. Màn hình hình chính

Giao diện chính của ứng dụng được thiết kế đơn giản, hỗ trợ người dùng nhập dữ liệu y tế và tạo bản tóm tắt nhanh chóng.

Chú thích:
- (1–3) Chọn nguồn dữ liệu đầu vào
  Người dùng lựa chọn cách cung cấp dữ liệu:
  1. Biểu tượng máy ảnh (1) – Chụp ảnh tài liệu y để hệ thống trích xuất văn bản bằng OCR.

2. Biểu tượng tài liệu (2) – Upload tài liệu văn bản trực tiếp.
3. Biểu tượng hình ảnh (3) – Tải ảnh có sẵn từ thiết bị và hệ thống trích xuất văn bản bằng OCR.

- (4) Chọn chế độ tóm tắt

Người dùng lựa chọn mức độ tóm tắt:

- Extract – Tóm tắt trích xuất, giữ nguyên câu quan trọng từ văn bản gốc (Hướng phát triển).
- Short – Tóm tắt ngắn gọn, cô đọng nội dung chính.
- TL;DR – Tóm tắt cực ngắn, cung cấp ý chính tổng quát.

- (5) Nhập nội dung

Văn bản sau khi OCR hoặc nội dung người dùng nhập thủ công sẽ hiển thị tại ô này. Người dùng có thể chỉnh sửa trước khi xử lý.

- (6) Thực hiện tóm tắt

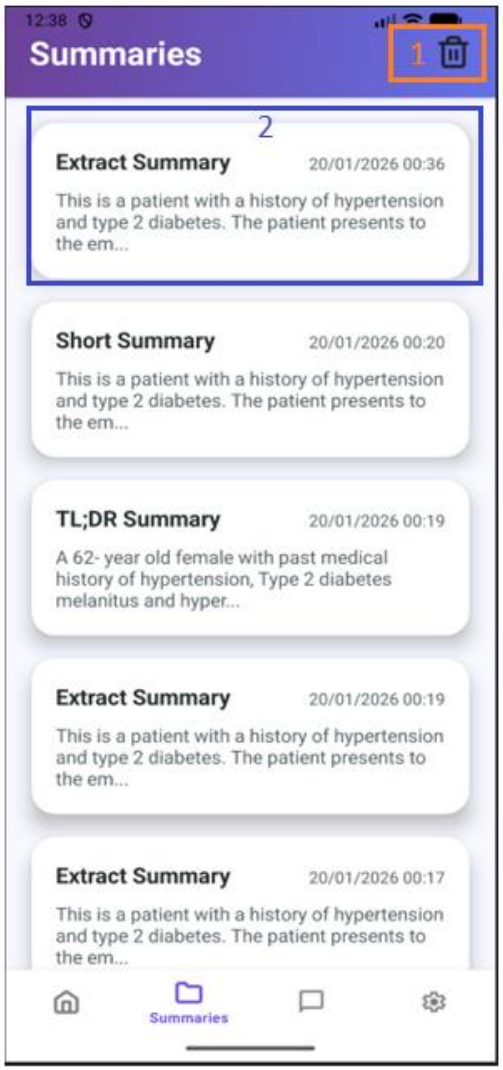
Nhấn nút APPLY để hệ thống AI xử lý và tạo bản tóm tắt dựa trên chế độ đã chọn.

- (7) Điều hướng chức năng

Thanh menu phía dưới cho phép truy cập các mục chính:

- Home – Màn hình tóm tắt chính
- Files – Quản lý tài liệu đã lưu
- Chat – Trao đổi với AI về nội dung y tế
- Settings – Cài đặt hệ thống

- Màn hình danh sách đã tóm tắt



Hình 3.6. Màn hình danh sách bản tóm tắt

Màn hình Summaries cho phép người dùng xem lại, quản lý và truy xuất các bản tóm tắt y tế đã được hệ thống tạo trước đó.

Chú thích:

(1) Biểu tượng thùng rác – Xóa dữ liệu

Nút ở góc trên bên phải cho phép người dùng:

- Xóa toàn bộ danh sách bản tóm tắt đã lưu
- Giải phóng bộ nhớ và làm mới dữ liệu lưu trữ

(2) Danh sách các bản tóm tắt

Mỗi mục trong danh sách đại diện cho một kết quả tóm tắt trước đó và bao gồm:

- Tiêu đề bản tóm tắt
- Thời gian tạo

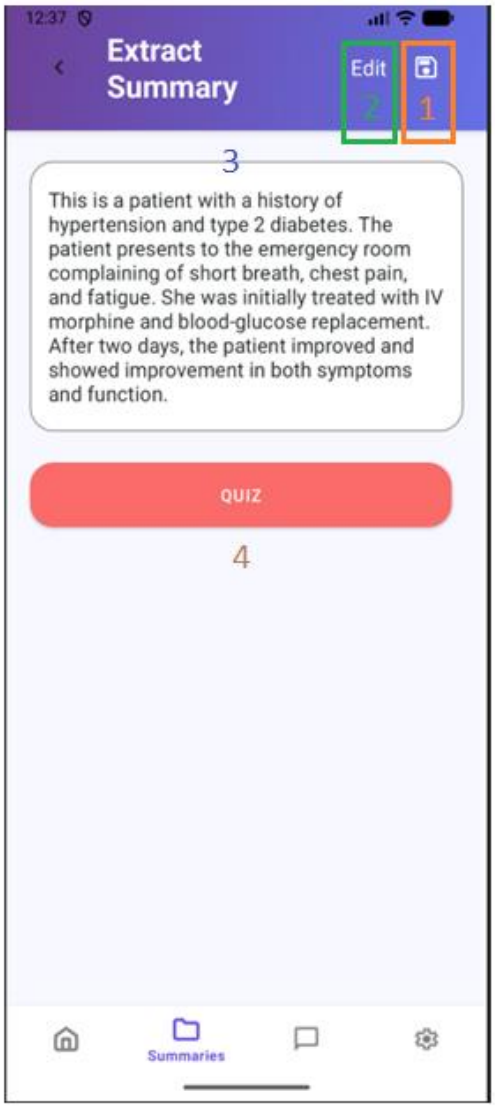
Hiển thị ngày và giờ hệ thống sinh bản tóm tắt, giúp theo dõi lịch sử xử lý.

- Nội dung rút gọn

  Hiển thị một phần nội dung để người dùng nhận diện nhanh tài liệu.

Người dùng có thể nhấn vào từng mục để xem chi tiết đầy đủ

- Màn hình chi tiết bản tóm tắt



Hình 3.7. Màn hình chi tiết bản tóm tắt

Màn hình này xuất hiện sau khi người dùng chọn một bản tóm tắt trong danh sách lịch sử. Chức năng chính là xem nội dung đầy đủ và thực hiện các thao tác liên quan.

Chú thích:

(1) Biểu tượng lưu

Nút ở góc trên bên phải cho phép:

- Lưu bản tóm tắt vào bộ nhớ thiết bị

(2) Nút Edit

Cho phép người dùng:

- Chỉnh sửa lại tiêu đề bản tóm tắt

(3) Nội dung bản tóm tắt

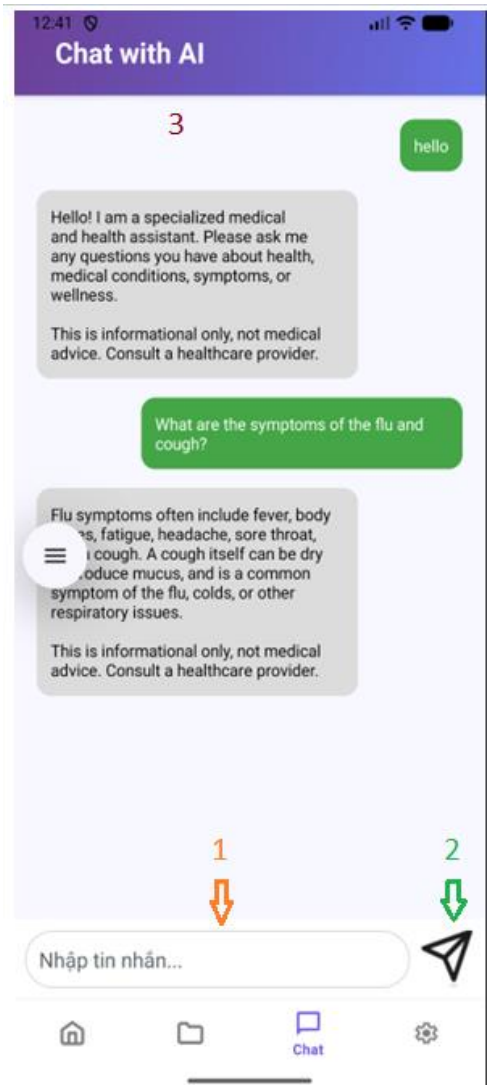
Hiển thị toàn bộ kết quả do hệ thống AI tạo ra dựa trên văn bản y tế đầu vào.
Người dùng có thể đọc, sao chép hoặc chỉnh sửa .

(4) Nút QUIZ (Hướng phát triển)

Kích hoạt chức năng tạo câu hỏi ôn tập từ nội dung tóm tắt.
Tính năng này hỗ trợ:

- Học tập kiến thức y khoa
- Kiểm tra mức độ hiểu nội dung văn bản

- Màn hình chat with ai:



Hình 3.8. Màn hình chat with AI

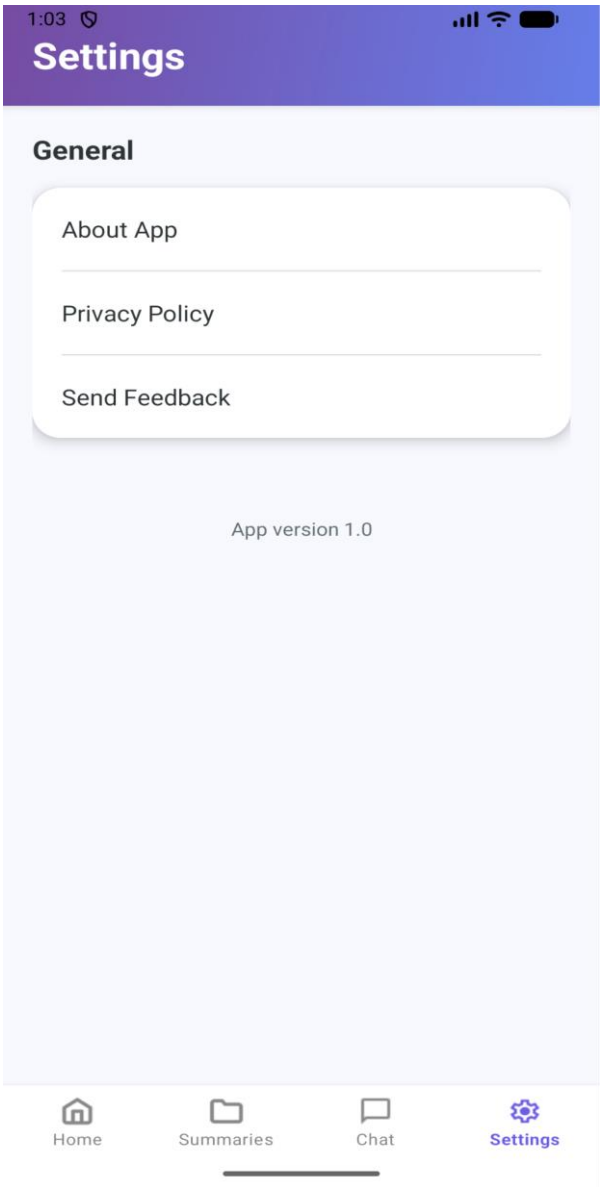
Chức năng Chat with AI cho phép người dùng tương tác trực tiếp với hệ thống trí tuệ nhân tạo để hỏi – đáp, giải thích hoặc khai thác thêm thông tin từ nội dung y tế đã tóm tắt.

Chú thích:
(1) Ô nhập nội dung
- Người dùng nhập câu hỏi hoặc yêu cầu tại đây
- Có thể hỏi về:
  - Giải thích bệnh lý
  - Ý nghĩa chỉ số xét nghiệm
  - Tóm tắt lại nội dung theo cách dễ hiểu hơn

(2) Nút gửi (Send)
- Gửi câu hỏi đến hệ thống AI
- Hệ thống xử lý và trả lời tự động

(3) Khu vực hiển thị hội thoại
- Hiển thị lịch sử trao đổi giữa người dùng và AI
- Tin nhắn người dùng và phản hồi từ AI được phân biệt bằng bố cục giao diện

- Màn hình cài đặt:



Hình 3.9. Màn hình cài đặt

Các chức năng này hiện tại mới ở mức giao diện (UI level) và chưa được tích hợp xử lý logic phía sau (backend/functional processing).

Màn hình Settings cung cấp thông tin về ứng dụng và hỗ trợ người dùng trong quá trình sử dụng.

Các mục chính bao gồm :

1. About App

Mục này hiển thị thông tin tổng quan về ứng dụng, bao gồm:

- Tên ứng dụng (*Medical Summary*)
- Phiên bản hiện tại
- Mục tiêu phát triển: hỗ trợ tóm tắt và khai thác thông tin y tế bằng AI

- Thông tin nhóm phát triển (nếu có)

2. Privacy Policy

Cung cấp chính sách bảo mật dữ liệu, bao gồm:

- Cách hệ thống xử lý dữ liệu văn bản và hình ảnh y tế
- Cam kết không chia sẻ thông tin cá nhân trái phép
- Quy định lưu trữ và bảo vệ dữ liệu người dùng

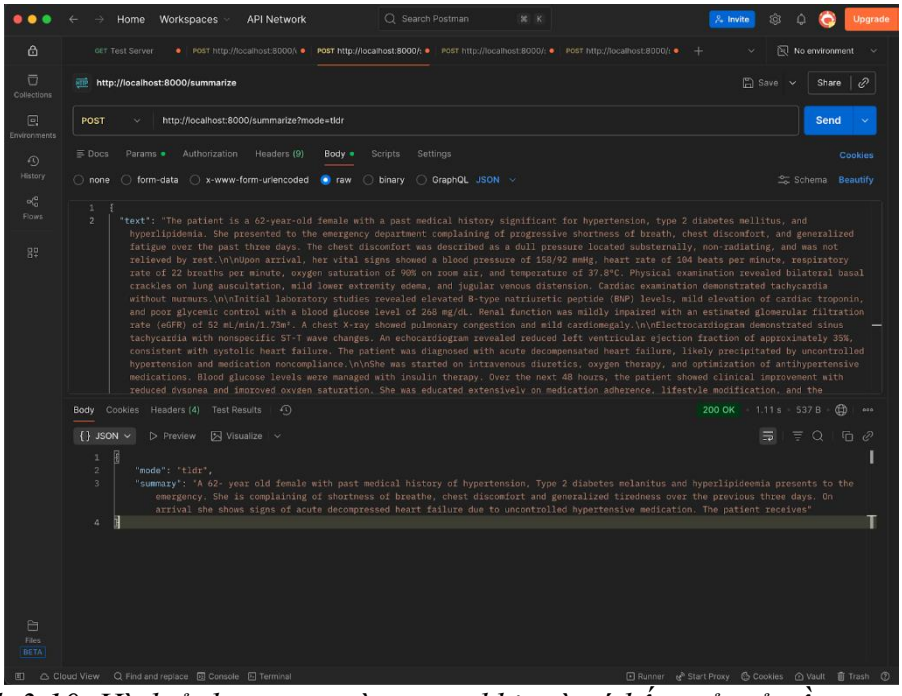
3. Send Feedback

Cho phép người dùng gửi góp ý hoặc báo lỗi:

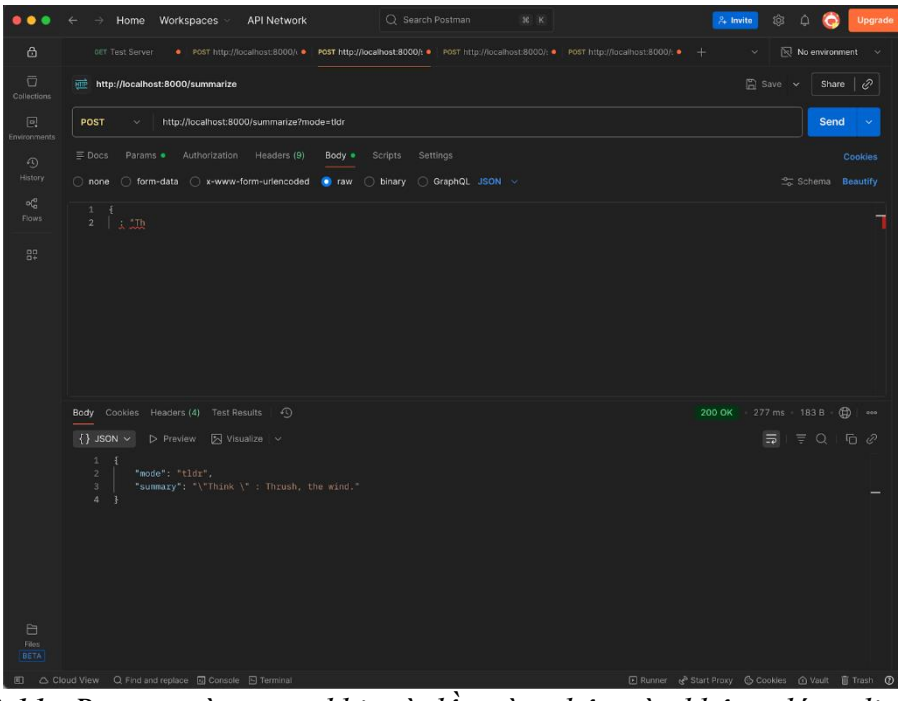
- Gửi phản hồi về chất lượng tóm tắt
- Báo lỗi hệ thống
- Đề xuất tính năng mới

Thông tin phản hồi có thể được gửi qua email hoặc biểu mẫu tích hợp.

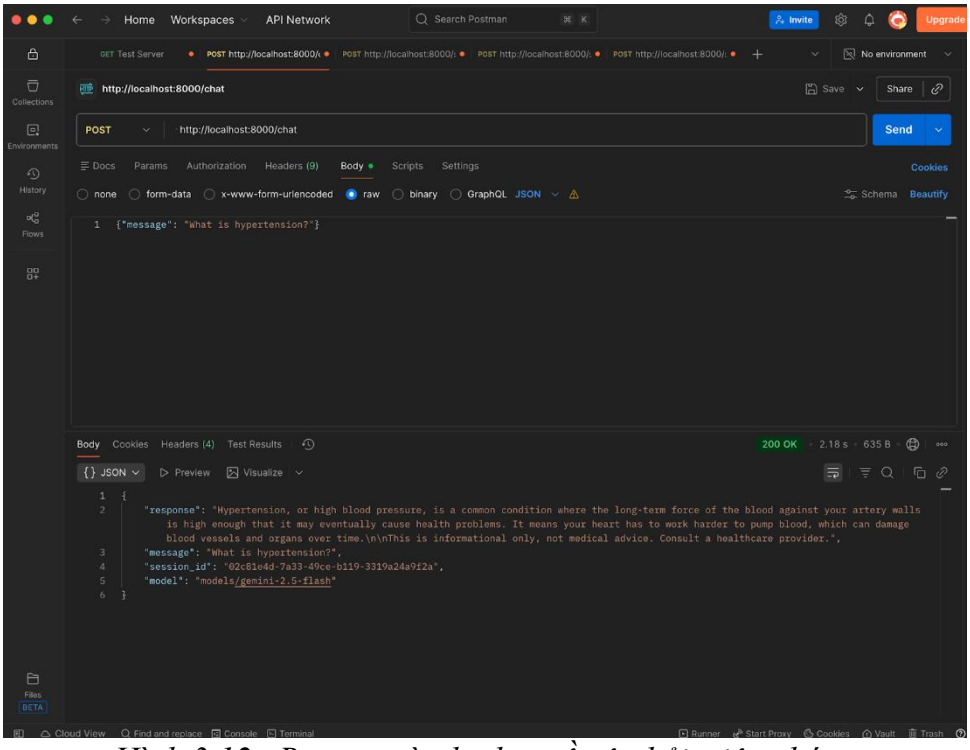
3.3. Kiểm thử



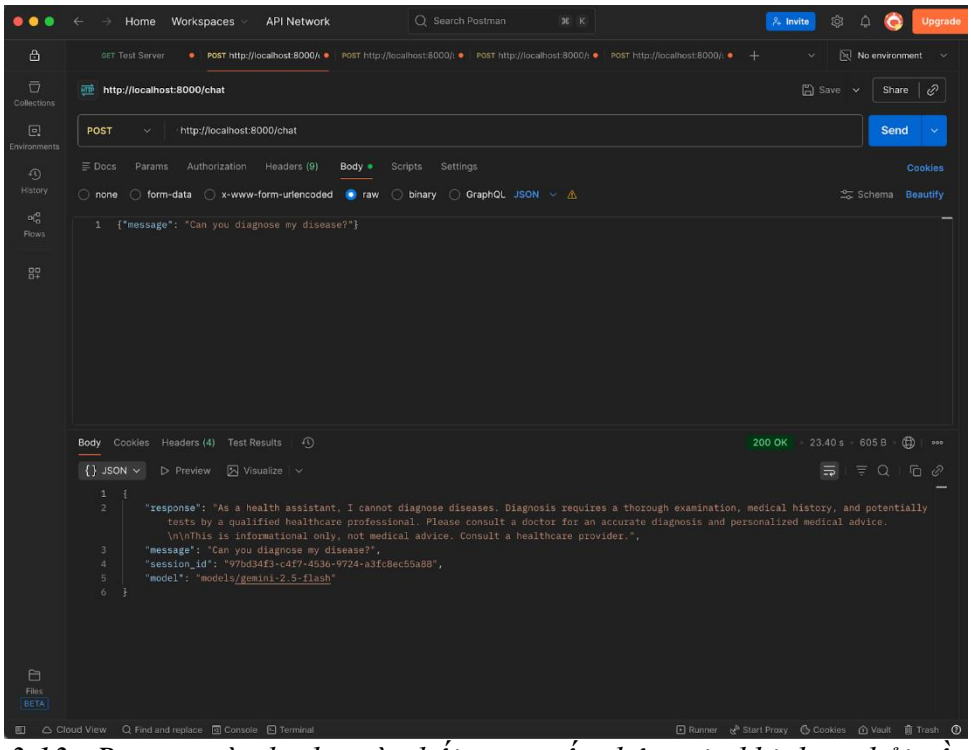
Hình 3.10. Hình ảnh request từ server khi mà có kết quả trả về status 200



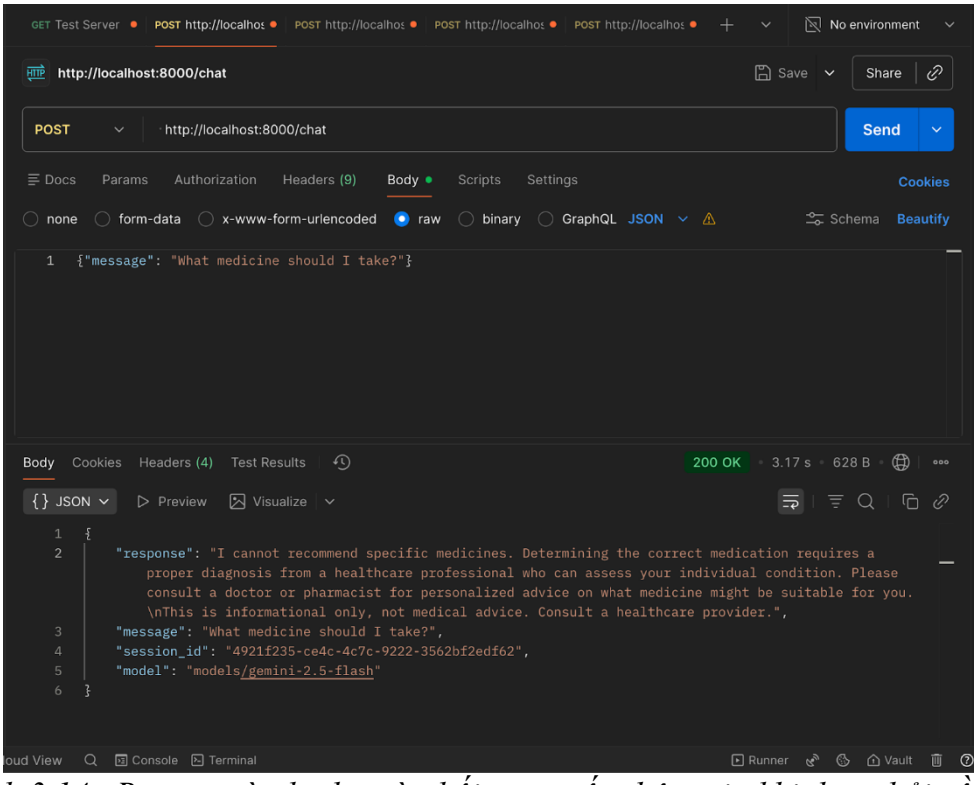
Hình 3.11. Request từ server khi mà đầu vào nhập vào không đúng định dạng



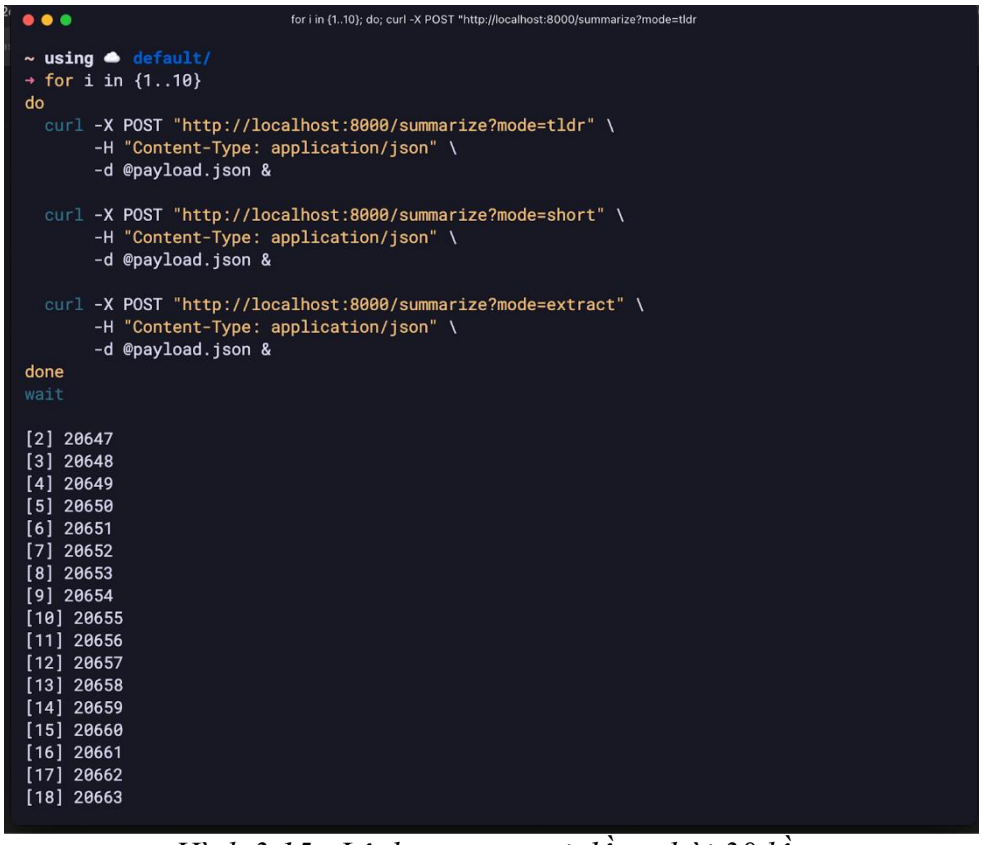
Hình 3.12. Request từ chatbot về câu hỏi triệu chứng



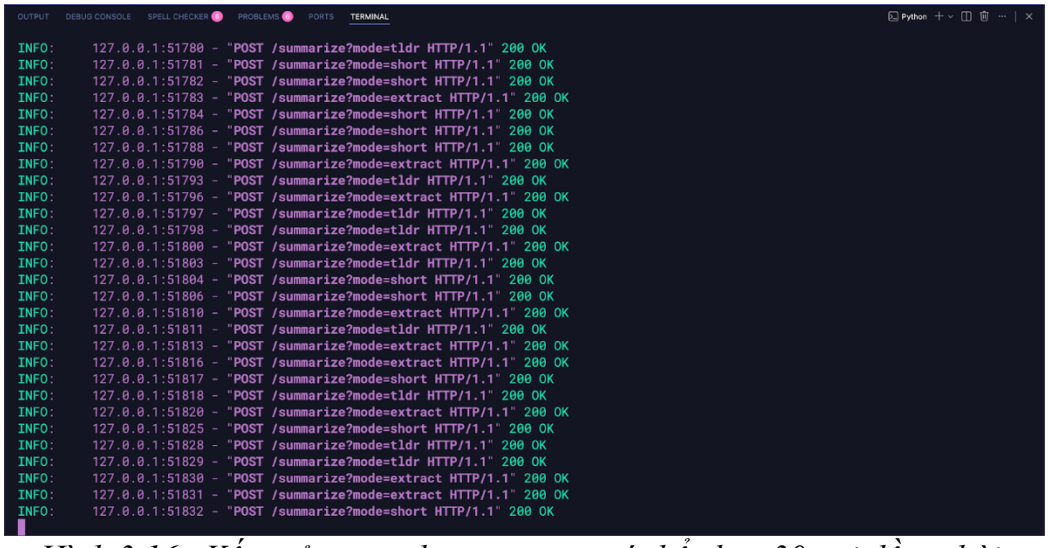
Hình 3.13. Request từ chatbot từ chối cung cấp thông tin khi được hỏi về chuẩn đoán bệnh



Hình 3.14. Request từ chatbot từ chối cung cấp thông tin khi được hỏi về nên uống thuốc gì và khuyên đi gặp bác sĩ



Hình 3.15. Lệnh request api đồng thời 30 lần



Hình 3.16. Kết quả server log ra server có thể chạy 30 api đồng thời

3.4. Đánh giá kết quả

- Ứng dụng đã được hoàn thiện ở mức cơ bản, đáp ứng các chức năng chính theo mục tiêu đề tài.

- Hệ thống cho phép người dùng nhập văn bản y tế và thực hiện tóm tắt nội dung bằng mô hình trí tuệ nhân tạo.

- Chatbot tư vấn sức khỏe hoạt động ổn định

- Giao diện người dùng được xây dựng bám sát thiết kế trên Figma, đảm bảo tính

trực quan và dễ sử dụng.

- Quá trình tích hợp giữa giao diện ứng dụng và backend diễn ra thông suốt, phản hồi tương đối nhanh.

- Kết quả tóm tắt đạt chất lượng phù hợp cho mục đích hỗ trợ tiếp cận thông tin và tiết kiệm thời gian đọc tài liệu.

3.5. Kết chương 3

Trong chương 3, quá trình xây dựng và hiện thực hóa hệ thống đã được trình bày chi tiết, bao gồm môi trường phát triển, công cụ lập trình và cách triển khai các chức năng chính của ứng dụng. Chương này đã mô tả việc tích hợp mô hình trí tuệ nhân tạo vào hệ thống, cũng như việc xây dựng giao diện người dùng trên nền tảng di động. Kết quả đạt được cho thấy hệ thống hoạt động đúng theo thiết kế đề ra, tạo nền tảng cho việc đánh giá, cải tiến và mở rộng ứng dụng trong các hướng phát triển tiếp theo.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết quả đạt được

Trong quá trình nghiên cứu tìm hiểu kiến thức, lý thuyết và triển khai ứng dụng đồ án đã đạt được những kết quả sau:

- Về mặt lý thuyết:

- Trong quá trình tìm hiểu và thực thi dự án đã giúp chúng em nắm vững và áp dụng được những kiến thức cơ bản để xây dựng nên một ứng dụng
- Rèn luyện kỹ năng phân tích nghiệp vụ, thiết kế hệ thống và xây dựng quy trình xử lý dữ liệu cho một ứng dụng thực tế.

- Về mặt ứng dụng:

- Xây dựng và triển khai thành công ứng dụng di động hỗ trợ tóm tắt văn bản y tế.
- Tích hợp mô hình trí tuệ nhân tạo vào hệ thống backend để xử lý yêu cầu tóm tắt từ người dùng.
- Thiết kế giao diện ứng dụng trực quan, bám sát bản thiết kế Figma và đảm bảo khả năng sử dụng cơ bản.
- Ứng dụng hoạt động ổn định, đáp ứng được mục tiêu hỗ trợ người dùng tiếp cận tài liệu y tế nhanh hơn.

2. Hạn chế

Ngoài những kết quả đã đạt được trong quá trình thực hiện đề tài. Hệ thống còn một số hạn chế như:

- Chất lượng tóm tắt phụ thuộc vào mô hình và dữ liệu huấn luyện, chưa được đánh giá sâu bằng các phương pháp đánh giá thủ công.

- Ứng dụng mới dừng lại ở mức hỗ trợ tóm tắt, chưa tích hợp các chức năng nâng cao như cá nhân hóa hoặc truy vấn thông tin chuyên sâu.

- Hiệu năng hệ thống có thể bị ảnh hưởng khi xử lý văn bản có độ dài lớn.

- Vì thời gian có hạn, kinh nghiệm thực tế chưa nhiều nên việc phân tích bài toán về cơ bản đã thực hiện tương đối đầy đủ, tuy nhiên chưa mô tả đầy đủ mọi khía cạnh của vấn đề.

3. Hướng phát triển

- Mở rộng tập dữ liệu huấn luyện để cải thiện chất lượng và độ chính xác của kết quả tóm tắt.

- Nghiên cứu tích hợp thêm các kỹ thuật đánh giá và kiểm chứng chất lượng đầu ra của mô hình.

- Phát triển thêm các chức năng như đặt câu hỏi theo nội dung tài liệu, highlight thông tin quan trọng hoặc hỗ trợ đa ngôn ngữ.

TÀI LIỆU THAM KHẢO

[1] Q. Xie et al., "A Survey for Biomedical Text Summarization: From Pre-trained to Large Language Models," arXiv:2304.08763, 2023. (Survey toàn diện về BTS từ PLM đến LLM, bao gồm preprocessing và challenges).

[2] N. M. Abdelaziz et al., "From Data to Insights: A Survey on Biomedical Text Summarization Approaches and Challenges," International Journal of Computers and Informatics, 2024-2025.

[3] Various works on clinical NLP preprocessing, e.g., from surveys in PMC and arXiv on biomedical entity preservation during normalization.

[4] A. Vaswani et al., "Attention is All You Need," Advances in Neural Information Processing Systems, 2017. (Nguồn gốc Transformer).

[5] Goldsack et al. (related works), "Faithful Medical Summarization via LLM," Nature Medicine và các paper 2023-2024 về clinical summarization.

[6] Y. Labrak et al., "BioMistral: A Collection of Open-Source Pretrained Language Models for Medical Domains," 2024 (hoặc các paper tương tự về fine-tuning cho biomedical).

[7] P. Szemraj, "long-t5-tglobal-base-16384-book-summary," Hugging Face Model Hub, 2022. (https://huggingface.co/pszemraj/long-t5-tglobal-base-16384-book-summary).

[8] Google Research, "LongT5: Efficient Text-to-Text Transformer for Long Sequences," arXiv preprint, 2021. (Paper gốc về LongT5 – https://arxiv.org/abs/2112.07916).

[9] Various works on long-document summarization, e.g., from surveys on LongT5 applications in technical domains.

[10] Falconsai/medical_summarization (T5 fine-tuned for medical), Hugging Face, tham khảo cho fine-tune y tế.

[11] Từ các paper về LongT5 trong medical evidence summarization (e.g., MedReview dataset fine-tune LongT5, 2024).

[12] Z. Huang et al., "A survey on biomedical automatic text summarization with large language models," Information Processing & Management, vol. 62, no. 5, 104216, 2025.

[13] N. M. Abdelaziz et al., "From Data to Insights: A Survey on Biomedical Text Summarization Approaches and Challenges," International Journal of Computers and Informatics, 2024.

[14] Various studies on entity hallucination in abstractive BTS, PMC, 2024-2025.

[15] Systematic review on LLM-based clinical summarization, JMIR, 2025.

[16] Works on entity preservation in medical text summarization, 2024.

[17] Hallucination mitigation techniques for faithful BTS, Nature Medicine-related papers, 2024.

[18] Surveys on long-document challenges and solutions in BTS, arXiv 2023-2025.

[19] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information*

Processing Systems, 2017, pp. 5998–6008.
[20] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5485–5551, 2020.
[21] Y. Liu *et al.*, "A Survey on Long-document Summarization: 2024 Update," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2024.
[22] M. Guo *et al.*, "LongT5: Efficient Text-to-Text Transformer for Long Sequences," in *Proc. of NAACL*, 2022, pp. 724–736.
[23] H. Zhang *et al.*, "Transient Global Attention: A Mechanism for Efficient Long-context Processing," *AI Communications*, vol. 37, no. 2, 2024.
[24] J. Wang, "The Evolution of Transformer Models for Long Form Summarization (2023-2025)," *CS Review*, 2025.
[25] T. Nguyen *et al.*, "Fine-tuning LongT5 for Medical Evidence Summarization on MedReview Dataset," *Journal of Biomedical Informatics*, vol. 142, p. 104386, 2023.
[26] S. Stan, "Evaluation of LongT5 on Biomedical Research Papers," *Hugging Face Technical Reports*, 2023. [Online]. Available: https://huggingface.co/Stancld/longt5-tglobal-large-16384-pubmed
[27] K. Miller, "Small but Mighty: Open Source Models vs GPT-4 in Medical Summarization," *Medical AI News*, 2024.
[28] R. Zhao *et al.*, "Robustness of Global-Local Attention in Domain-Specific NLP," *Nature Machine Intelligence (Preprint)*, 2025.
[29] P. Szemraj, "BookSum as a Pre-training Objective for Long-form Content," *Medium AI Blog & Paper Series*, 2022.
[30] E. J. Hu *et al.*, "LoRA: Low-Rank Adaptation of Large Language Models," in *Proc. of ICLR*, 2022 (Updated 2024).
[31] T. Dettmers *et al.*, "QLoRA: Efficient Finetuning of Quantized LLMs," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023.
[32] L. Huang *et al.*, "Adapting Large Language Models for Clinical Summarization: LoRA vs. Full Fine-tuning," *Journal of Biomedical Informatics*, vol. 148, p. 104523, Dec. 2023.
[33] G. Liu *et al.*, "MOELoRA: An Aspect-adaptive Parameter-efficient Fine-tuning Framework for Medical Applications," *arXiv preprint arXiv:2310.18339*, 2024.
[34] J. Smith, "State of PEFT in Domain-Specific NLP: A 2025 Review," *AI in Medicine*, vol. 12, no. 1, pp. 45-62, 2025.
[35] T. M. Nguyen, "Challenges in Medical AI Development for Emerging Economies: A Study on Resource-efficient Models," *Vietnam Journal of Computer Science*, 2024.
[36] ccdv/pubmed-summarization, Hugging Face Datasets.
[37] T. Zhang et al., "BERTScore: Evaluating Text Generation with BERT," ICLR, 2020 (và các ứng dụng trong BTS).
[38] Kotlin Documentation, JetBrains & Google, 2023-2025.
[39] Android Developers Guide: Kotlin for Android, Google, 2024.

[40] MVVM Architecture on Android, Google Jetpack, 2024.
[41] Coroutines and Flow in Android, Google, 2024.
[42] SwiftUI Overview, Apple Developer Documentation, 2024.
[43] Declarative UI with SwiftUI, Apple WWDC, 2023-2025.
[44] Combine Framework Integration with SwiftUI, Apple, 2024.
[45] Migrating from UIKit to SwiftUI, Apple Developer, 2024.
[46] FastAPI Documentation, Tiange, 2024.
[47] S. Ramírez, "FastAPI: High Performance, Easy to Learn, Fast to Code, Ready for Production," 2018-2025.
[48] Hugging Face Transformers Integration with FastAPI, Hugging Face Docs, 2025.
[49] RESTful API Design Guidelines, Microsoft & Google, 2024.
[50] JWT Authentication in FastAPI, FastAPI Security, 2024.
[51] MVC Pattern in Modern Web Frameworks, ACM Surveys, 2023.
[52] Room Persistence Library, Android Jetpack, Google, 2024.
[53] Core Data Framework, Apple Developer Documentation, 2024.