ШИНЭ МОНГОЛ ТЕХНОЛОГИЙН КОЛЛЕЖ

КОМПЬЮТЕРЫН УХААНЫ ИНЖЕНЕРИЙН ТЭНХИМ

Оюутны код: s21c076b
Оюутны овог нэр: Батцэнгэл АНАР

RAG технологид тулгуурласан локал файл хайлтын ухаалаг туслагч

/ТӨГСӨЛТИЙН СУДАЛГААНЫ АЖИЛ/

Удирдагч багш Б.Батчулуун
Гүйцэтгэсэн оюутан Б.Анар

Улаанбаатар хот
2025 он

ШИНЭ МОНГОЛ ТЕХНОЛОГИЙН КОЛЛЕЖ

КОМПЬЮТЕРЫН УХААНЫ ИНЖЕНЕРИЙН ТЭНХИМ

Төгсөлтийн судалгааны ажил

RAG технологид тулгуурласан локал файл хайлтын ухаалаг туслагч

Гүйцэтгэгч: Б.Анар

Удирдагч: Б.Батчулуун

Улаанбаатар хот

2025 он

Хураангуй

Энэхүү төгсөлтийн судалгааны ажлын зорилго нь өөрийн сүлжээнд холбогдоогүй орчинд хадгалагдаж буй өгөгдлөөс хиймэл оюун ухаанд суурилсан RAG (Retrieval-Augmented Generation) аргаар мэдээлэл хайх систем боловсруулах явдал юм. Сургуулийн дотоод сүлжээ эсвэл компьютерын диск дотор хадгалагдаж буй баримтууд, бичвэрүүдийг хэрэглэгчийн асуултад үндэслэн автоматаар хайж, хамгийн тохирох хариултыг гаргах системийг бүтээхээр төлөвлөж байна.

Орчин үед мэдээлэл асар хурдтай өсөж байгаа боловч байгууллагын дотоод мэдээллийг үр дүнтэй ашиглахад хүндрэл гардаг. Иймээс энэхүү судалгаагаар дотоод орчинд ажиллах, өгөгдлийн аюулгүй байдлыг хамгаалсан хиймэл оюун ухааны хайлтын систем боловсруулах замаар уг асуудлыг шийдвэрлэхийг зорьж байна.

Төслийн хүрээнд FAISS вектор хайлтын сан, Hugging Face-ийн LLM хэлний загвар, болон LangChain буюу урьдчилан хөгжүүлсэн системийг ашиглан энэхүү системийн туршилтын хувилбарыг боловсруулах бөгөөд эхний шатанд өгөгдлийн бүтцийг тодорхойлох, бичвэрүүдийг вектор хэлбэрт хувиргах, дараагийн шатанд хэрэглэгчийн асуултанд тулгуурлан холбогдох баримт илрүүлэх ба хариулт үүсгэх системийг хэрэгжүүлэхээр төлөвлөж байгаа.

Агуулга

Товчилсон үгийн жагсаалт

1. **RAG** – Retrieval-Augmented Generation — Хайлтад суурилсан хариулт үүсгэх систем
2. **FAISS** – Facebook AI Similarity Search — Вектор ижил төстэй байдлаар хайлт хийх сан
3. **Hugging Face** – Хиймэл оюун ухааны загварууд, датасет хуваалцах нээлттэй платформ
4. **LLM** – Large Language Model — Том хэлний загвар
5. **UI** – User Interface — Хэрэглэгчийн интерфэйс
6. **PDF** – Portable Document Format — Баримт бичгийн олон улсын стандарт формат
7. **JSON** – JavaScript Object Notation — Өгөгдөл солилцох стандарт бүтэцтэй формат.
8. **AI** – Artificial Intelligence — Хиймэл оюун ухаан.
9. **API** – Application Programming Interface — Програм хооронд харилцах холболт
10. **VScode** – Visual Studio Code — Кодыг ажлуулдаг өргөтгөлтэй программ

Хүснэгтийн жагсаалт

Зургийн жагсаалт

1 Ажлын төлөвлөгөө

Хүснэгт1 Судалгааны ажлын төлөвлөгөө

Хүснэгт 1: Судалгааны ажлын төлөвлөгөө

№	Төлөвлөгөө	Эхлэх	Дуусах	Явц
1	Онол, арга зүйн материалуудыг судлах	2025.10.15	2025.10.31	
2	Системийн ерөнхий төлөвлөгөө боловсруулах	2025.11.1	2025.11.15	
3	Диск доторх хайлт болон нэвтэрч чадах файлуудыг судлах	2025.11.15	2025.11.30	
4	ТСА үзлэг 1	2025.12.2		
5	компьютерийн дискээс автоматаар хайх модуль турших	2025.12.3	2025.12.15	
6	Хариултын системийг сайжруулах, алдаа засварлах	2025.12.15	2025.12.28	
7	Векторчлох болон хайлт хариултын системийг сайжруулах турших	2026.1.2	2026.1.15	
8	Хэрэглэгчийн UI болгох туршилт болон бичвэрийн засвар	2026.1.15	2026.1.31	
9	UI хийн холбох туршилт хийх	2026.2.1	2026.2.15	
10	Туршилтын демо бэлтгэх	2026.2.15	2026.2.28	
11	ТСА үзлэг 2	2026.3.1		
12	Хариулт хайлт оновчтой болгох	2026.3.2	2026.3.15	
13	Хэрэглэгчийн туршилт	2026.3.15	2026.3.31	
14	Судалгааны бичвэрийг боловсруулж дуусгах	2026.4.1	2026.4.15	
15	Туршилтын үр дүнг нэгтгэх	2026.4.15	2026.4.29	
16	ТСА урьдчилсан хамгаалалт	2026.4.30		
17	Урьдчилсан хамгаалалтын зөвлөгөө авч сайжруулах	2026.5.1	2026.5.15	
18	ТСА хамгаалалтанд бэлэн болох	2026.5.15	2026.5.30	
19	ТСА үндсэн хамгаалалт	2026.5.31		

2 Удиртгал

2.1 Үндэслэл, ач холбогдол

Одоо үед мэдээллийн технологийн хөгжлийн хурд нэмэгдэж, их хэмжээний өгөгдлийг богино хугацаанд боловсруулах, хайх, дүн шинжилгээ хийх шаардлага улам өсөж байна. Гэвч компьютерын дотоод дискэнд хадгалагдсан файлуудаас шаардлагатай мэдээллийг гараар хайх нь цаг их шаарддаг, алдаа гарах магадлал өндөр үйл явц юм.

APQC-ийн нийтлэг судалгаагаар (жишээ нь 2023 оны KM Benchmark):

- Мэдлэгийн ажилтнууд ажлынхаа 20–30%-ийг зөвхөн “мэдээлэл хайх эсвэл дахин бүтээхэд” зарцуулдаг. → Энэ нь өдөрт дунджаар 1.8–2 цаг орчим алдаж байна гэсэн үг.
- Мэдээлэл амархан олддог байгууллагууд (good knowledge systems бүхий) энэ хугацааг 50% хүртэл багасгаж чаддаг.
- Хайлтын үр дүн 1 минут тутамд амжилттай гарч байвал, тухайн байгууллагын productivity-д 15–25% өсөлт гардаг гэж тооцдог.

Мөн зарим компаниуд хуучин хөгжүүлж байсан код олох ойлгоход хугацаа шаарддаг ба шинээр ирсэн ажилтан өмнөх ажилтны кодыг ойлгохгүй байх тохиолдол ч мөн байдаг. Тиймээс хэрэглэгчийн асуултад үндэслэн дотоод дискнээс мэдээлэл автоматаар хайж, холбогдох хариулт гаргах систем боловсруулах нь мэдээлэл боловсруулах ажлыг хялбарчлах, хугацаа хэмнэх, ажлын бүтээмжийг дээшлүүлэхэд чухал ач холбогдолтой.

2.2 Зорилго, зорилт

Энэ судалгааны зорилго нь компьютерын дотоод дискэнд хадгалагдсан бичвэрийн өгөгдлийг боловсруулан, хэрэглэгчийн асуултад тохирох мэдээллийг автоматаар илрүүлж, холбогдох хариултыг гаргаж чаддаг Retrieval-Augmented Generation (RAG) архитектурт суурилсан хариултын системийг боловсруулан ажлын цаг хэмнэх үр бүтээмжтэй байдлыг нэмэгдүүлэх юм.

Зорилт:

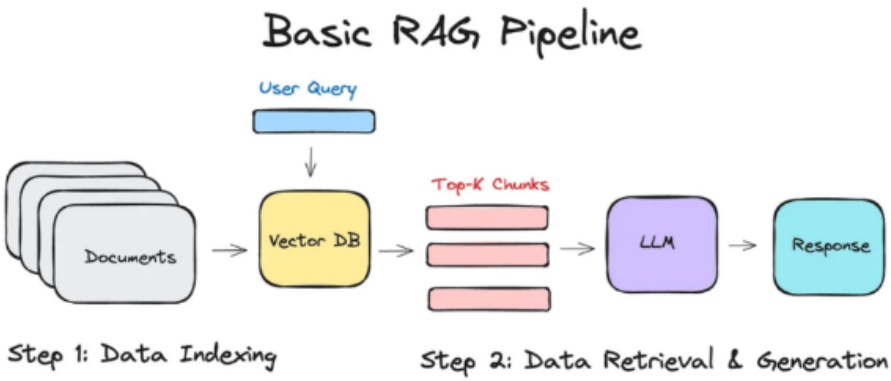
- Локал дискнээс өгөгдөл унших, индексжүүлэх, вектор хэлбэрт хөрвүүлэх процессыг RAG системд тохируулах
- Hugging Face pipeline ашиглан хэрэглэгчийн асуултад үндэслэн утгачилсан (context-aware) хариу үүсгэх
- FAISS ашиглан өгөгдлийн хайлтын хурд, нарийвчлалыг сайжруулах
- Хайлтын системийг туршиж оновчтой байдлыг нэмэгдүүлэх

- Цаашид хөгжүүлж чадвал UI-тай буюу хэрэглэгчид хэрэглэхэд амар болгон программ болгон харуулах

3 Судалгааны сэдвийн онол, өнөөгийн түвшин

3.1 Онолын хэсэг

Зураг1 RAG үндсэн бүтэц



Зураг 1: RAG үндсэн бүтэц

RAG систем нь дараах гурван үндсэн бүрэлдэхүүнтэй:
Хайлт хийх хэсэг (Retriever)

- Энэ хэсэгт бүх баримтыг вектор хэлбэрт хөрвүүлэн хадгалдаг.
- Хэрэглэгчийн асуулт орж ирэхэд хамгийн холбоотой гэсэн баримтыг хурдан олж татна.
- Ашигладаг технологи: FAISS, Pinecone, Semantic Searchгэх мэт.
- Үр дүн: LLM-ын үндсэн сургагдсан мэдлэгээр хязгаарлагдахгүй шинэ эх сурвалжаас мэдээлэл ашиглаж чадна.

Хариу үүсгэх хэсэг:(Generator/ LLM)

- Хайлтын хэсгээс татсан баримтад үндэслэн хариу гаргана.
- Ингэснээр хариу нь зөвхөн LLM-ийн зохиох функцийн хариултаар бус шинэ баримттай мэдээллээс хариулт үүсгэнэ.
- Ашигладаг загварууд:T5, GPT, FLAN-T5,ollamагэх мэт.

Асуулт ангилах, уялдуулах хэсэг (Query Classifier / Router)

- Хэрэглэгчийн асуултыг төрөл, агуулгаар нь ангилж, тохирох баримт руу чиглүүлнэ.

Ажлах дараалал:

- Documents → Баримт бичгүүдийг вектор хэлбэрт хөрвүүлэн хадгална
- User Query → Хэрэглэгч асуулт оруулна
- TOP-K Chunks → Хамгийн холбоотой K ширхэг хэсгүүдийг сонгож авна
- LLM → OpenAI GPT, Gemini зэрэг том хэлний модель хариуг боловсруулна
- Response → Эцсийн хариуг хэрэглэгчид харуулна.

Үндсэн кодны хувьд python ашиглан visual studio code программ дээр хөгжүүлэлт хийсэн. Python нь олон төрлийн үндсэн функц, сан ихтэй хөгжүүлэлт хийхэд амар тул сонгосон.

FAISS (Facebook AI Similarity Search) том хэмжээний өгөгдөл дотор ойролцоо векторуудыг хурдан хайдаг. Санах ой бага зарцуулдаг ба туршихад хялбар сүлжээнд холбогдоогүй байсан ч ажиллах боломжтой.

3.2 Ижил төрлийн судалгаа

Experimental Study on RetrievalAugmented Generation: Engineering and Evaluation of a Custom RAG system for OpenDomain QA (University ofPadova, 2024/2025)

Evaluation Metrics for Retrieval Augmented Generation in the Scientific Domain (2025)

A RetrievalAugmented Generation Framework for Academic Literature Navigation in Data Science (arXiv 2024)

Эдгээр судалгаа нь адилхан RAG систем ашигласан боловч зөвхөн pdf, dock гэсэн хязгаарлагдмал өргөтгөлтэй бичиг баримт дээр ажилладаг. Эсвэл сүлжээтэй газар үүлэн сервер дээрээс хайлт хийдгээрээ ялгаатай.

4 Судалгааны арга зүй

Энэхүү судалгаа нь хэрэглээний судалгаа бөгөөд энэ нь хүмүүсийн хувийн болон компанийн файл, бичиг баримттай харьцах, хайх хугацааг багасган илүү үр дүнтэй хурдан ажиллахад хэрэг болоход чиглэсэн. Чанарын судалгаа ашиглан ямар хиймэл оюун ухааны модел илүү хариулт гаргаж байгааг үнэлсэн.

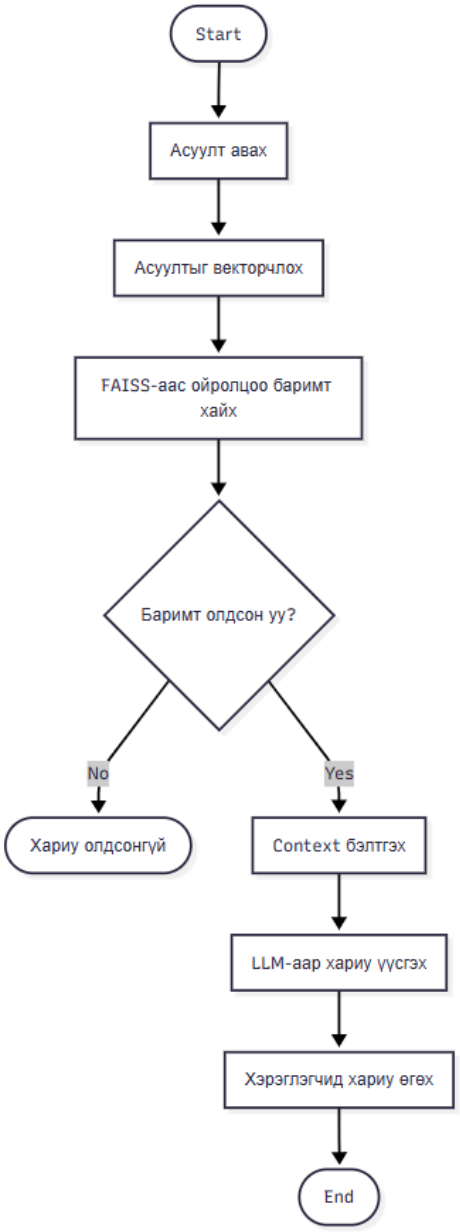
Дотоод системээс хайлт хийх учир хандаж чадах бичвэрийн өргөтгөлийг судалсан.

Хүснэгт2 Хандаж чадах өргөтгөлтэй файлууд

Хүснэгт 2: Хандаж чадах өргөтгөлтэй файлууд

№	Төрөл	Өргөтгөл	Ашиглах сан	Тайлбар
1	Текст файл	`. txt`	Built-in (open)	Энгийн текст файл
2	Markdown	`.md`	Built-in (open)	Форматтай текст файл
3	Log файл	`.log`	Built-in (open)	Системийн лог файл
4	PDF баримт	`.pdf`	PyPDF2	Portable Document Format
5	Word баримт	`.docx`	python-docx	Microsoft Word (шинэ)
6	Word баримт	`.doc`	python-docx	Microsoft Word (хуучин)
7	Excel хүснэгт	`.csv`	pandas	Comma-Separated Values
8	JSON өгөгдөл	`.json`	json (built-in)	JavaScript Object Notation
9	JSON Lines	`.jsonl`	json (built-in)	Мөр тус бүр JSON
10	PowerPoint	`.pptx`	python-pptx	Microsoft PowerPoint (шинэ)
11	PowerPoint	`. ppt`	python-pptx	Microsoft PowerPoint (хуучин)

Үндсэн ажиллагааг зураг 2 харуулсанчлан хийхээр төлөвлөсөн тул эхлээд өгөгдөл авч түүндээ хариулж чадаж байгаа үгүйг шалган мөн тохирох LLM-ын загваруудыг туршиж үзэн сонгосон.



Зураг 2: Activation диаграм

FAISSвектор дата бааз ашигланembedding model(векторжуулалтын загвар)-оорвектор болгон хувиргаж хурдан хайлт хийх боломжтой болгосон.

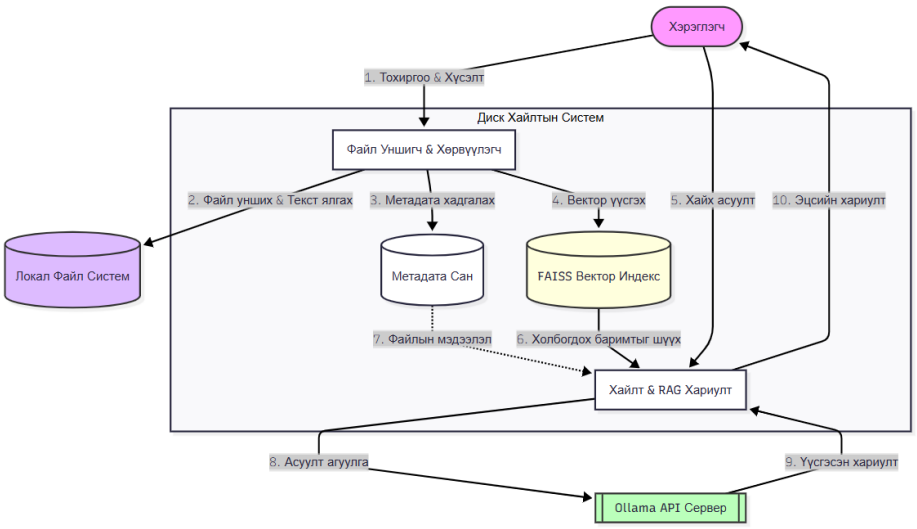
LLM-уудыг вектор болгосон өгөгдөл дундаас хэрэглэгч асуулт асуухад хариулж чадаж байгаа үгүйг судлахаар нэг нэгээр нь туршсан ба туршихдаа гурван шалгуур тавьсан. Хариулах буюу бичвэр үүсгэх хурд, асуулт ойлгох чадвар, хариулт үүсгэх гэсэн гурван шалгуур. Үр дүнд зураг 3 дээр харагдаж буй загварууд ашиглахад амар ба хариулт зөв гаргаж байгаа гэж дүгнэсэн.

Хүснэгт3 LLM-уудын туршсан үр дүн

Хүснэгт 3: LLM-уудын туршсан үр дүн

Загвар	Шалтгаан
FLAN-T5-Base, Qwen2:0.5b	Хөнгөн, хурдан (8GB RAM)
Qwen2:1.5b, Llama3.2:3b	Чанар, хурдны тэнцвэр (16GB RAM)
Qwen2:7b, Llama3:8b	Хамгийн сайн чанар (32GB+ RAM)
Qwen2	Азийн хэлний дэмжлэг (Монгол хэлэнд)
Llama3, DeepSeek	Код сайн ойлгодог
FLAN-T5, Qwen2	Q&A-д сургагдсан

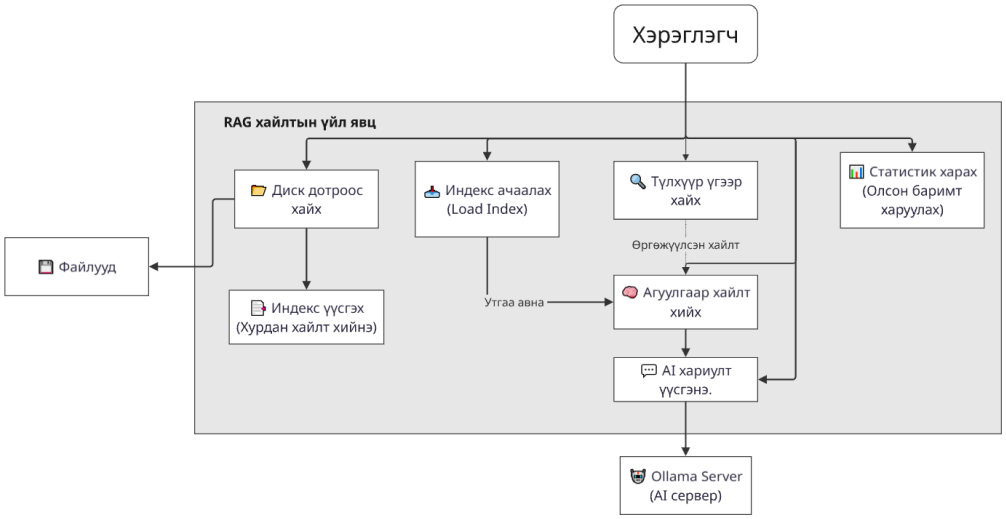
Зураг3 LLM-уудын туршсан үр дүн



Зураг 3: LLM-уудын туршсан үр дүн

Энэ судалгааны хувьд олон хүнд хүртээмжтэй байлгахын тулд багтаамж их эзэлдгүй LLM буюу qwen2-ыг сонгосон ба кодыг өгөгдлийн урсгал диаграммаар харуулбал зураг 4-т харуулж байна.

Зураг4 Dataflow диаграмм



Зураг 4: Dataflow диаграмм

Зураг5 User case диаграмм

[User case диаграмм]
image7.png

Зураг 5: User case диаграмм