-Các nhánh chính của AI :

+Machine Learning : dạy cho máy tính học từ dữ liệu để đưa ra dự đoán hoặc quyết định

Ví dụ : dự đoán giá nhà, giá cổ phiếu, hệ thống gợi ý

+Deep learning (thuộc ML) : sử dụng mạng noron nhiều lớp để đưa ra kết quả

Ví dụ : nhận dạng khuôn mặt, xử lý ảnh

+GenAI: tạo ra nội dung mới như văn bản, hình ảnh,âm thanh, … bằng các kỹ thuật như fine-tuning, prompt, RAG, ..

Ví dụ : chatbot, content marketing, sáng tác nghệ thuật

+NLP : giúp máy tính hiểu và sinh ngôn ngữ tự nhiên như con người bằng các kỹ thuật như tokenization, …

Ví dụ: Dịch tự động, trợ lý ảo

+ Computer Vision : giúp máy hiểu nội dung từ ảnh

Ví dụ : Nhận diện khuôn mặt, …

+Robotics : Kết hợp AI và phần cứng (robot công nghiệp,..)

+Expert Systems : hệ thống chẩn đoán bệnh,…

+Planning : giải các bài toán lên lịch, đường đi ngắn nhất ,…

-Lộ trình làm chatbot :

https://chatgpt.com/share/687e1689-8130-800a-91d5-5c67ef07ad3d

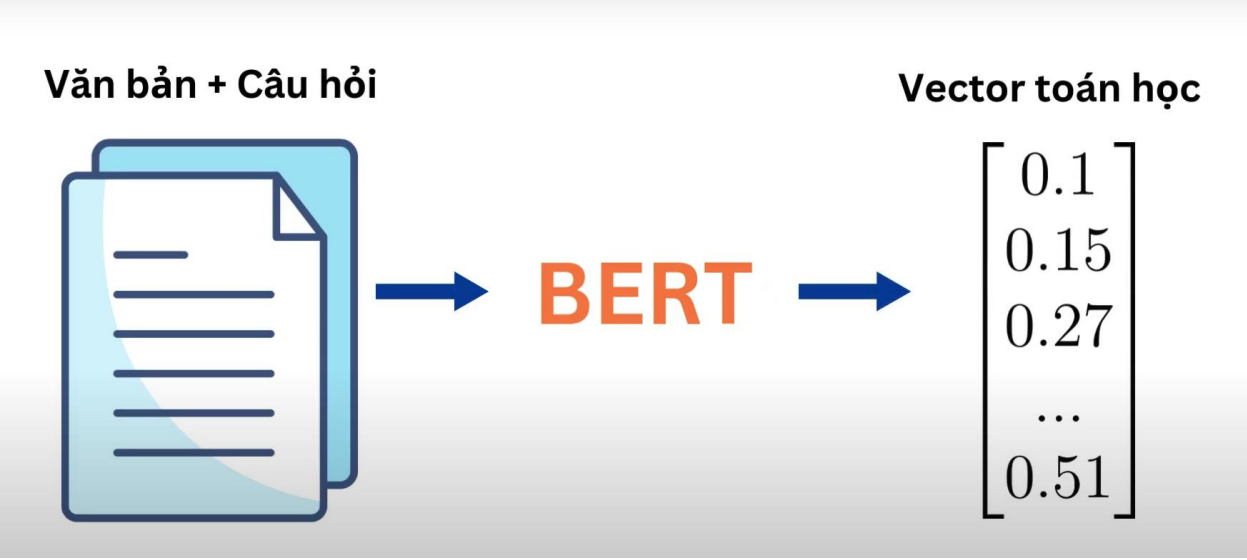
-Tìm hiểu về RAG và dify (tự động hóa workflow, có 2 loại là chatflow và workflow) :

+Các mô hình ngôn ngữ lớn thường có limited knowledge, việc re-training dữ liệu mới cũng khiến tốn nhiều chi phí

⇨ Khi gắng hỏi thì hay bị hallucination (ảo giác kiến thức)

+RAG gồm hai thành phần :

+ Retrieve : thu thập các văn bản có liên quan đến câu hỏi của người dùng



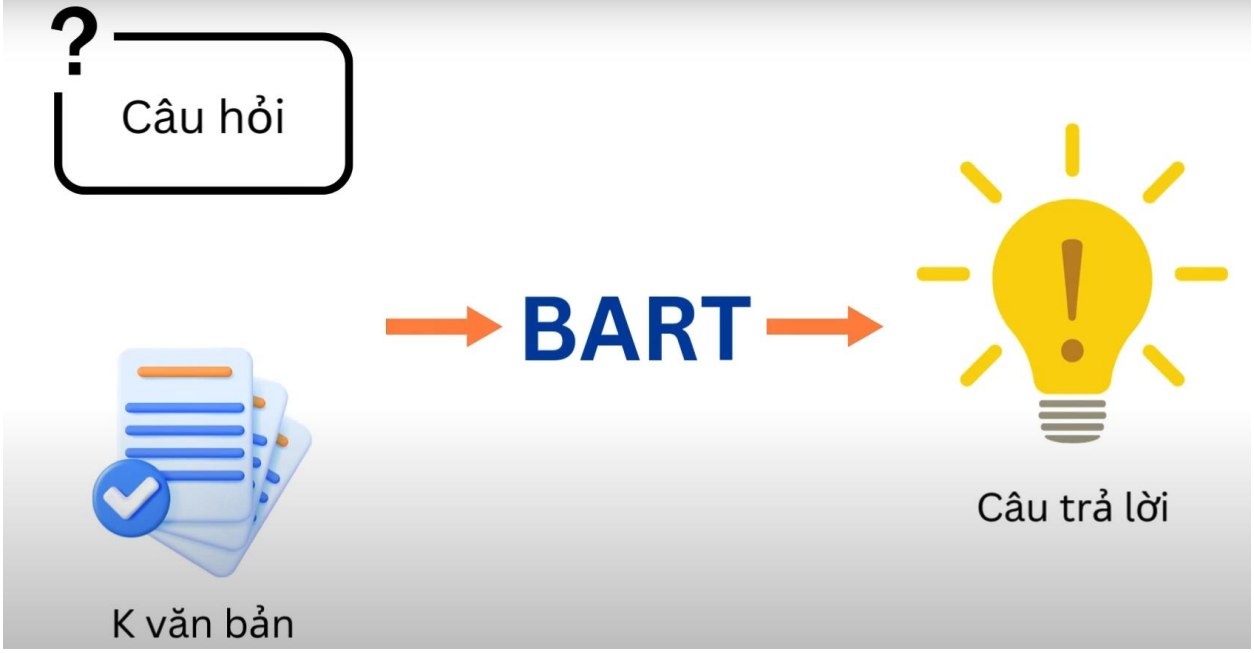
Tìm K văn bản sao cho

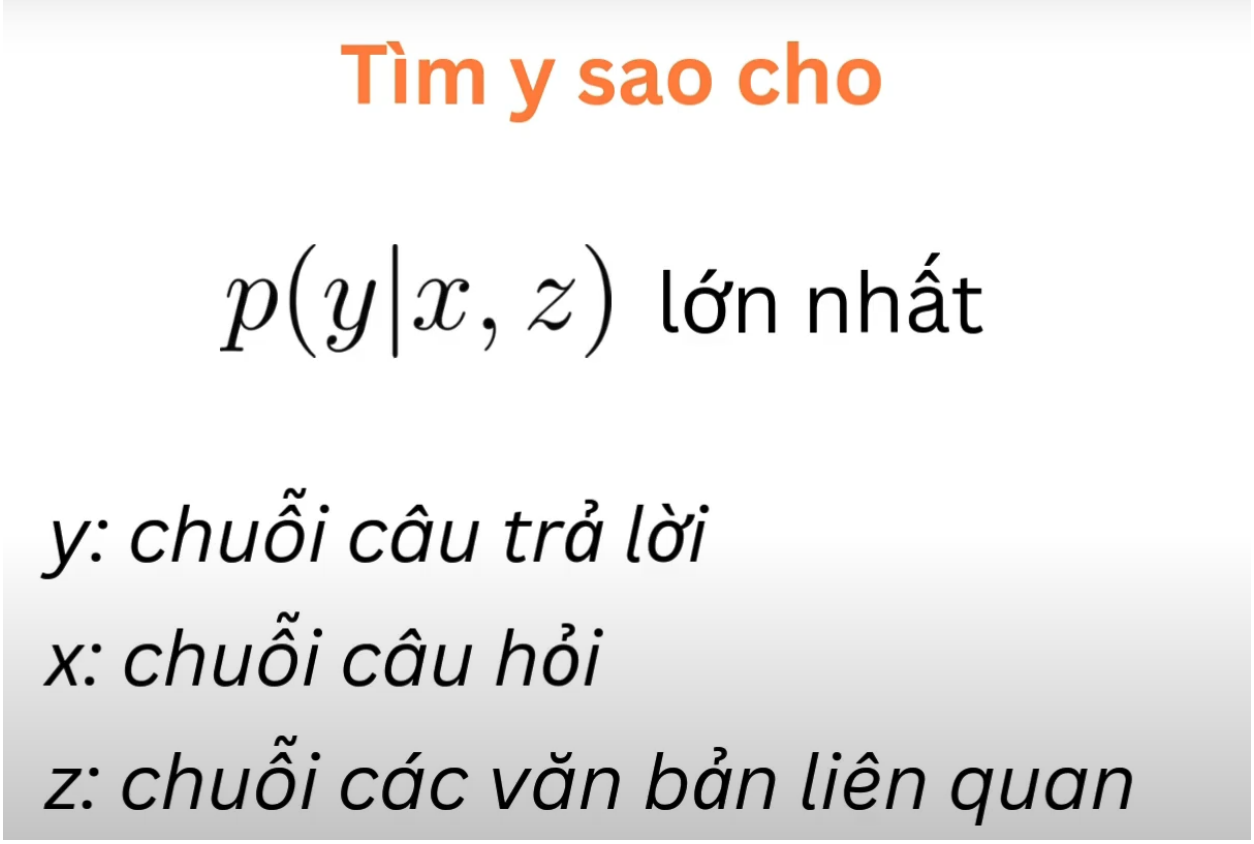
$$d^T q \text{ lớn nhất}$$

d: vector mã hoá văn bản

q: vector mã hoá câu hỏi

+ Generate : Tạo sinh ngôn ngữ sử dụng các văn bản liên quan


Câu hỏi
BART
Câu trả lời
K văn bản


Tìm y sao cho
$p(y|x,z)$ lớn nhất
y: chuỗi câu trả lời
x: chuỗi câu hỏi
z: chuỗi các văn bản liên quan

+ Ví dụ :

```python
from chatgpt import answer_question, answer_question_with_context

QUESTION = ("How many planets are there in the solar system and"
            "what are they called")

CONTEXT = ("Scientists at NASA just found another planet in the solar"
           "system called Earth-2. It is now recognised as a new planet.")

if __name__ == "__main__":
    # print(f"Answer without new context: {answer_question(QUESTION)}")
    print(f"Answer with new context: {answer_question_with_context(CONTEXT, QUESTION)}.")
```

\- ナレッジベース:Tính năng cơ sở kiến thức của Dify sử dụng hệ thống RAG hỗ trợ AI tạo sinh, cho phép người dùng dễ dàng tạo và quản lý dữ liệu của riêng mình, và kiến thức họ tạo ra có thể được sử dụng làm ngữ cảnh tìm kiếm. Các tệp dữ liệu tương ứng bao gồm nội dung dài và dữ liệu có cấu trúc, đồng thời có thể được liên kết đến các trang web, Notion, Github và cơ sở dữ liệu.

-チャンク：Một khối là một phần nhỏ dữ liệu độc quyền được RAG sử dụng và Dify tự động chia các đoạn văn bản hoặc dữ liệu dài thành các khối để xử lý, giúp hệ thống hiểu thông tin một cách hiệu quả và tạo ra các phản hồi phù hợp.

ーChuẩn bị data source. Có thểở nhiều dạng :

Long text content (TXT, Markdown, DOCX, HTML, JSONL, or even PDF files)

Structured data (CSV, Excel, etc.)

Online data source(Web pages, Notion, etc.)

hoặc có thể kết nối với knowledge base bên ngoài

-Các bước tạo knowledge base :

+import text data

+chọn chunk mode, có thể xem trước kết quả

   +sau khi chunking, sử dung phương pháp top-k retrieval để lấy các chunk có độ liên quan cao nhất

   +Cần làm sạch dữ lieu tránh dữ liệu không cần thiết (dòng trống, các ký tự vô nghĩa, ...)

   +Chiến lược làm sạch: Loại bỏ các ký tự không cần thiết, khoảng trắng thừa, dòng trống.Xóa URL và địa chỉ email không liên quan.

****General mode:****

+Chia theo chunk delimiter (ký tự phân cách)

+maximum length, overlapping length

+Quy tắc tiền xử lý: thay thế khoảng trắng liên tiếp

+Sử dung Q\&A : tạo từng chunk chứa câu hỏi và câu trả lời từ nội dung da tai len

****Parent-Child mode:****

Sử dụng cấu trúc hai tầng: Parent Chunk (Chunk Cha,ngữ cảnh) và Child Chunk (Chunk Con, thông tin chi tiết).

+Parent chunk : chế độ paragraph, chế độ full document (vượt quá 10k ký tự sẽ bị bỏ bớt)

+child chunk giống general chunk nhưng dc tạo từ chunk cha

=>cung cấp ngữ cảnh toàn diện hơn, phù hợp cho tài liệu dài

+CHọn cấu hình chỉ mục và cấu hình retrieval

+Chờ cho tiến trình hoàn than

-Tiền xử lý và làm sạch :ETL (trích xuất, chuyển đổi và tải)

+Dify ETL và Unstructured ETL

-Embedding : chuyển doi cac văn bản rời rạc sang vector

-Metadata

-Setting the indexing method:

*High-quality indexing: sử dung dc Q\&A va các search

*Econimical indexing:sử dung 10 từ khóa trên mỗi chunk de truy xuat, truy xuất theo inverted index dựa vào các từ khóa

+Sử dung mô hình embedding de chuyen cac chunk van ban thanh vector so

-Setting truy xuất dữ liệu (retrieval)

*Vector Search :Vector hóa câu hỏi của người dùng để tạo vector truy vấn, sau đó so sánh với các vector chunk trong cơ sở kiến thức để tìm các chunk gần nhất.

+Rerank model: sắp xếp text chunk duoc tra ve boi vector search dựa vào mối quan hệ ngữ nghĩa

+TopK: số chunk dc retrieval từ query của người dùng

+Score Threshold: ngưỡng diem toi thieu de truy xuat

+TopK và score threshold chỉ được dùng khi enable rerank

*Full-text search: Lập chỉ mục tất cả các thuật ngữ trong tài liệu, cho phép người dùng truy vấn bất kỳ thuật ngữ nào và trả về các đoạn văn bản (text chunk) có chứa các thuật ngữ đó.

*Hybrid search: thực hiện dong thoi ca hai phuong phap,Kết quả được sắp xếp lại để chọn các chunk phù hợp nhất.

+Weight settings : cân chỉnh do uu tien giua Semantic Search (truy xuat ngu nghia) va Keyword search(truy xuat tu khoa)

+So sánh hai phương pháp :

Semantic Value of 1 This activates only the semantic search mode. Utilizing embedding models, even if the exact terms from the query do not appear in the knowledge base, the search can delve deeper by calculating vector distances, thus returning relevant content. Additionally, when dealing with multilingual content, semantic search can capture meaning across different languages, providing more accurate cross-language search results.

Keyword Value of 1 This activates only the keyword search mode. It performs a full match against the input text in the knowledge base, suitable for scenarios where the user knows the exact information or terminology. This approach consumes fewer computational resources and is ideal for quick searches within a large document knowledge base.

-Kiểm tra truy xuất dữ liệu :

Suggested Steps for Retrieval Testing:

Design and organize test cases/test question sets covering common user questions.

Choose an appropriate retrieval strategy: vector search/full-text search/hybrid search. For the pros and cons of different retrieval methods, please refer to the extended reading Retrieval-Augmented Generation (RAG).

Debug the number of retrieval segments (TopK) and the recall score threshold (Score). Choose appropriate parameter combinations based on the application scenario, including the quality of the documents themselves.

How to Configure TopK Value and Retrieval Threshold (Score)

TopK represents the maximum number of retrieval chunks when sorted in descending order of similarity scores. A smaller TopK value will recall fewer segments, which may result in incomplete recall of relevant texts; a larger TopK value will recall more segments, which may result in recalling segments with lower semantic relevance, reducing the quality of LLM responses.

The retrieval threshold (Score) represents the minimum similarity score allowed for recall segments. A smaller recall score will retrieval more segments, which may result in recalling less relevant segments; a larger recall score threshold will recall fewer segments, and if too large, may result in missing relevant segments.

-Xem thêm metadata và kiểm tra truy xuất dữ liệu