Bui Quang Manh @buiquangmanh Theo dõi
5.8K 241 41

Đã đăng vào khoảng 5 giờ trước - 8 phút đọc
15 0 0

Tổng quan Search Engine và Vector Database [Part 2]

KhaiButDauXuan ContentCreator

Trong bài viết trước , mình đã giới thiệu về các khái niệm như **vector search, vector database, search engine etc** và ví dụ qua một công cụ cloud-based search engine là Azure Cognitive Search. Như đã đề cập trong phần trước, Azure Cognitive Search cung cấp các hình thức tìm kiếm sau:

- Azure Full-Text Search
- Azure Vector Search
- Azure Hybrid Search

Trong bài viết trước, mình và các bạn đã điểm qua cách hoạt động của loại hình **Full-Text Search**. Hôm nay chúng ta điểm qua tiếp các hình thức còn lại nhé

C. Azure Cognitive Search

II. Azure Vector Search

1. Cách hoạt động của Azure Vector Search

Khi **indexing**, Azure Search sẽ lấy vector embedding và sử dụng thuật toán nearest neighbors algorithm để đặt các vector có ý nghĩa tương tự hoặc liên quan với nhau gần với nhau. Các vector này có thể tạo ra bằng cách sử dụng embedding model của Open AI, Azure OpenAI hoặc các custom model. Bạn cũng có thể sử dụng các phương pháp encoding đơn giản để chuyển đầu vào thành dạng vector. Kho lưu trữ tập hợp các vector mã hóa đó được gọi là **embedding space**

Khi **truy vấn (query)**, bạn cần tiến hành mã hóa input từ user về dưới dạng vector, sau đó sẽ gửi truy vấn tìm vector tương tự với input vector trong kho lưu trữ.

2. Thuật toán Nearest neighbors search

Như đã trình bày bên trên, search engines cần tìm vector trong không gian **embedding space** gần nhất với vector của input query. Và thuật toán này được gọi là **Nearest neighbor search**. Để tối ưu thời gian, độ trễ , thông lượng và bộ nhớ trong quá trình tính toán độ tương tự giữa các vector, Azure AI Search hỗ trợ hai thuật toán:

- **Hierarchical Nagigable Small World (HNSW)**:
    - Tổ chức dữ liệu nhiều chiều dưới dạng cấu trúc đồ thị phân cấp vì các vector giống nhau được phân cụm gần nhau nên việc đi tìm kiếm tương đương việc tìm cụm gần nhất với vector truy vấn,
    - Tìm kiếm nhanh, đáp ứng với số lượng điểm tìm kiếm lớn.
    - Tất cả dữ liệu đều được lưu trữ trên bộ nhớ để đảm bảo khả năng truy cập nhanh do đó thuật toán này sử dụng dung lượng **vector index**
- **Exhaustive K-nearest neighbors (KNN)**:
    - Tính toán độ tương đồng giữa vector cần truy vấn với toàn bộ các điểm dữ liệu.
    - Phù hợp với bộ dữ liệu nhỏ
    - Không lưu dữ liệu trên bộ nhớ
    - Không dùng dung lượng vector index

III. Hybrid search

Hybrid search là kết hợp của cả full-text search và vector search

2. Cấu trúc câu truy vấn của Hybrid Search

- **search** chỉ định câu truy vấn dành cho full text search
- **vectors** chỉ định dành cho truy vấn vectors
- **select** chỉ định trường nào được trả về trong kết quả
- **filters**: chỉ định tiêu chuẩn khác cần phải có ví dụ khoảng cách địa lý, nhiệt độ, etc
- **facets**
- **queryType=semantic** kích hoạt semantic ranking

D. Cách tính score của các hình thức tìm kiếm

Như vậy, các bạn đã đi qua các hình thức tìm kiếm được hỗ trợ của dịch vụ Azure. Như các bạn cũng biết, search engine chọn ra kết quả tốt nhất dựa trên một số điểm của mỗi kết quả được trả về. Sau đây chúng ta cùng tìm hiểu số điểm đấy được thực hiện tính toán như nào nhé.

I. Full-Text Search

1. Scoring algorithms

Score trong mỗi kết quả của hình thức tìm kiếm full-text search được tính bằng công thức thuật toán TF/IDF được viết tắt bởi cụm từ term frequency-inverse document frequency. Các bạn tìm hiểu thêm thuật toán TF/IDF sâu thêm ở đây nhé. Thuật toán này ưu tiên các từ có tần suất ít trong văn bản hơn các từ phổ biến, tăng khả năng hiệu quả việc tìm kiếm

Azure Full Text Search sử dụng thuật toán **BM25 ranking** cũng là một thuật toán bắt nguồn từ TD/IDF và điểm số này thể hiện qua giá trị **@search.score** ở kết quả trả ra.

2. Cách giải pháp nâng cao hiệu suất tìm kiếm

2.1. Scoring profiles

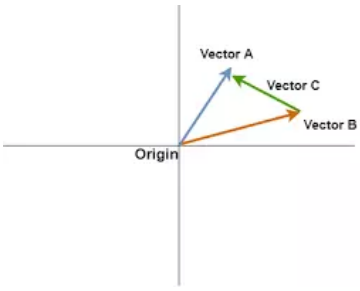
Xây dựng một bộ luật ưu tiên matching ở một số trường cụ thể hơn các trường khác. Ví dụ bạn tìm kiếm thông tin ở trường **nội dung** và **tiêu đề**, thì việc tìm kiếm được từ khóa tương ứng ở trường tiêu đề đóng góp điểm số cao hơn so với việc tìm kiếm được thông tin ở trường nội dung.

Hướng dẫn việc tạo *profile* được Azure hướng dẫn ở đây

2.2. Term boosting

Term boosting là một tập hợp các điều kiện để thực hiện tìm kiếm dựa trên cây truy vấn mà mình có đề cập ở phần trước. . Ví dụ bạn muốn tìm tất cả các từ có bắt đầu bằng air-condition*, hoặc kết thúc có cụm từ hoặc từ nào đó vân vân

II. Vector Search



Với Vector Search, chúng ta có các chỉ số chính là độ đo khoảng cách giữa các vector

- **cosine**: đo góc giữa hai vector, không bị ảnh hưởng bởi sự khác nhau của độ dài vector
- **dotProduct**: đo khoảng cách giữa hai cặp vector và góc giữa chúng
- **Euclidean**: đo khoảng cách giữa hai cặp vector tương đương l2-norm

III. Hybrid Search

Thuật toán Reciprocal Rank Fusion (RRF) tính ra score cuối cùng dựa trên đầu vào là kết quả đã được sắp xếp từ nhiều thuật toán khác. Đó cũng là lý do chúng được sử dụng trong phương thức Hybrid Search.

RRF quan tâm vị trí của các items trong danh sách ban đầu và đưa điểm cao hơn cho những item ở thứ hạng cao ở kết quả của nhiều phương thức.

- **Bước 1**: Lấy kết quả được xếp hạng từ nhiều các phương thức tìm kiếm khác nhua một cách song song
- **Bước 2:** Với mỗi document trong danh sách kết quả ở bước 1, RRF gắn score cho mỗi document bằng cách score = 1 / (rank + k) với rank là ví trí của document đó trong danh sách, k là một hằng số chỉ định và mặc định là 60
- **Bước 3**: Tính tổng RRF Score cho mỗi document.

Chú ý chỉ những trường được đánh dấu là **searchable** hoặc **searchField** mới được dùng cho scoring.

Tạm kết

Thời gian đọc cũng hơi dài rồi, hy vọng các nội dung bài hôm nay sẽ mang lại nhiều kiến thức bổ ích cho mọi người. Trong phần tiếp theo, chúng ta sẽ tìm hiểu sâu hơn về search engine, các biện pháp cải thiện hiệu suất mô hình nhé. Cảm ơn các bạn đã dành thời gian đọc và hẹn gặp lại trong phần tiếp theo.

Tài liệu tham khảo

- Azure AI Search documentation

#Search Engine | Azure Cognitive Search | Vector Search | hybrid search

Bài viết liên quan

Bài viết khác từ Bui Quang Manh

631 0 0 7

Tổng quan Search Engine và Vector Database [Part 1]

Bui Quang Manh

11 phút đọc

763 3 0 10

Tối ưu quá trình huấn luyện với tf.data API

Bui Quang Manh

7 phút đọc

181 0 2 3

Graph execution và Eager execution trong Tensorflow

Bui Quang Manh

7 phút đọc

215 0 0 3

SVTR NET - Lời giải hoàn hảo cho bài toán OCR ?

Bui Quang Manh

11 phút đọc

3.1K 9 5 28

Bình luận

Đăng nhập để bình luận