Carnegie Mellon University

AI Semantic Search on StackOverflow

Danny Chen

Agenda

- What is Semantic Search?
- Understanding the problem
- Data Preprocessing
- Model fine-tuning
- Using Cross Encoder to Re-rank results
- Evaluation and Future Plans

How often do you use Google Search?

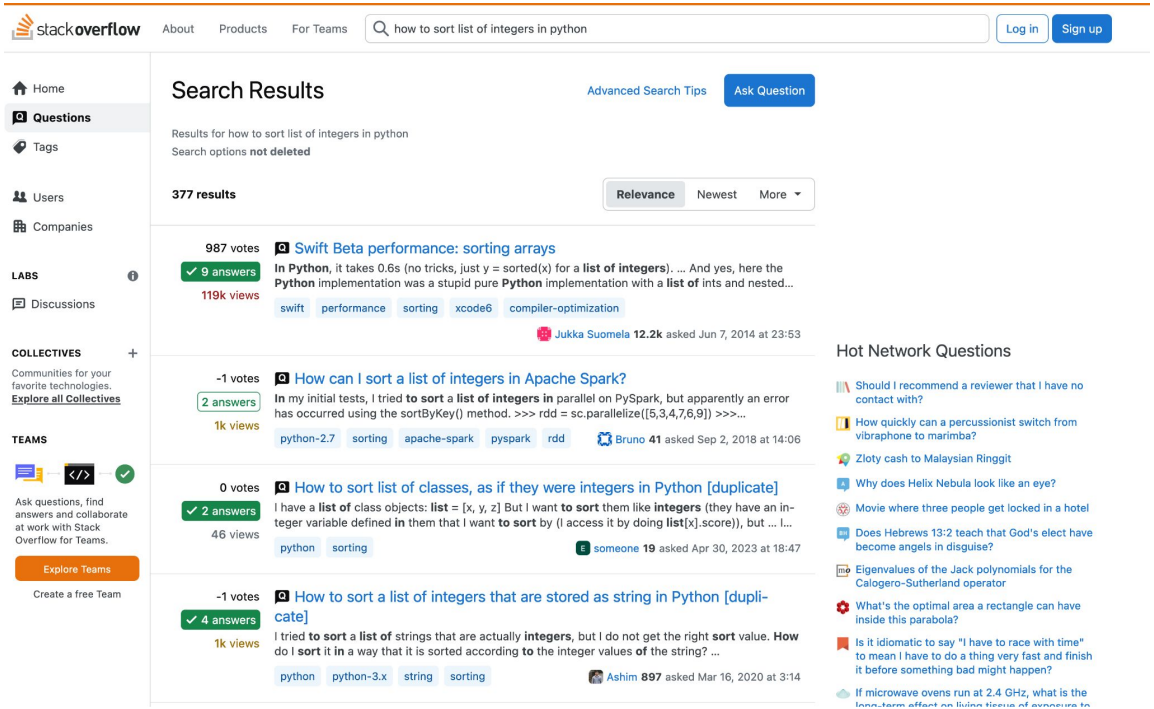
Did you know Google had advanced search operators?

Do you use it?

Search operator	What it does	Example
" "	Search for results that mention a word or phrase.	"steve jobs"
OR	Search for results related to X or Y.	jobs OR gates
|	Same as OR:	jobs | gates
AND	Search for results related to X and Y.	jobs AND gates
–	Search for results that don't mention a word or phrase.	jobs -apple
*	Wildcard matching any word or phrase.	steve * apple
( )	Group multiple searches.	(ipad OR iphone) apple
define:	Search for the definition of a word or phrase.	define:entrepreneur

Carnegie Mellon University

Challenges with Traditional Search

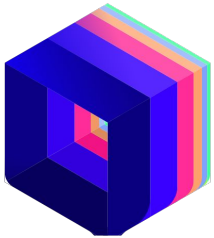


- Keyword dependence
- Lack of ability to understand word context
- Requires advanced search syntax to get good results

Carnegie Mellon University

What is Semantic Search?

“Focus on the search, not the syntax”

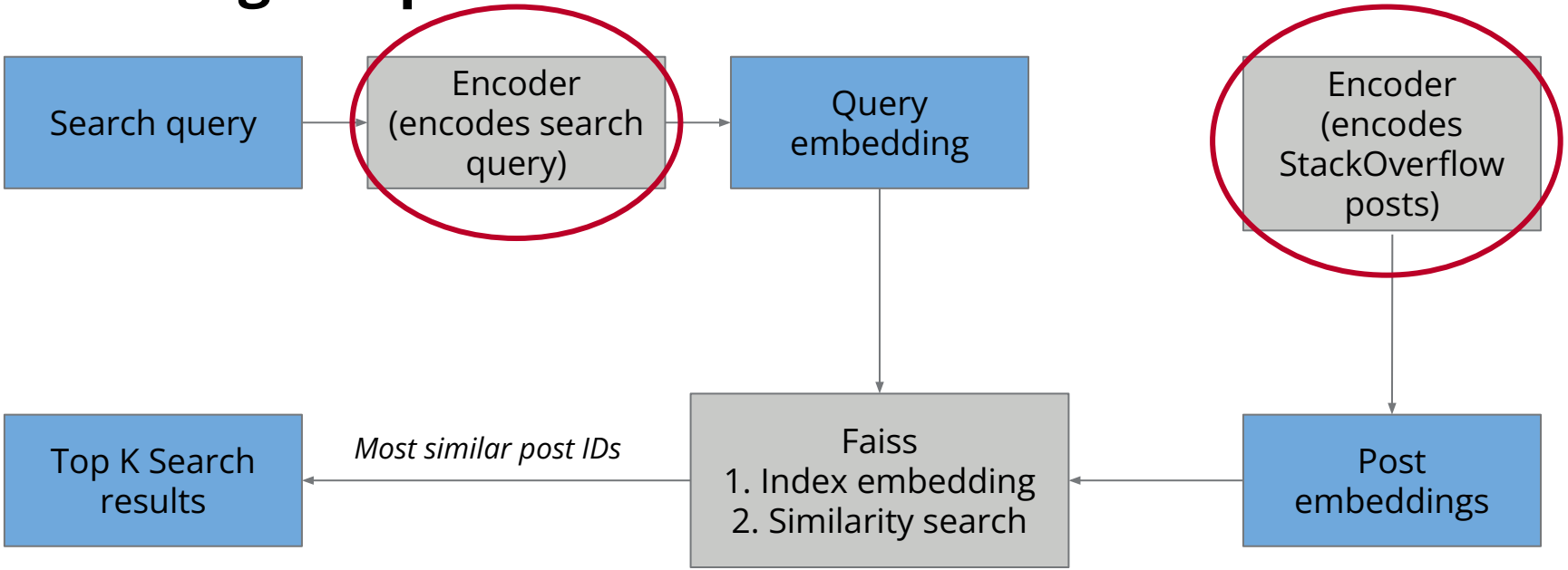


VectorDB

Carnegie Mellon University

Understanding the Problem

Modeling the problem

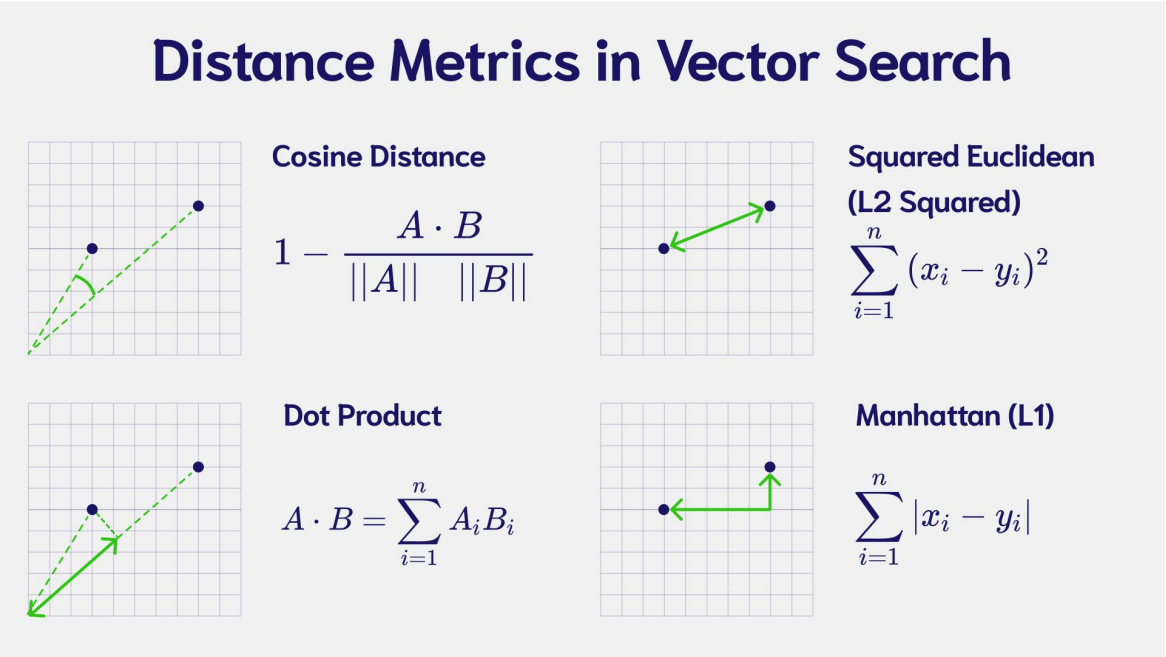


Symmetric vs Asymmetric Search

Symmetric Semantic Search - user query and the entries in the corpus are about the same length. You could potentially flip the query and entries in the corpus.

Asymmetric Semantic Search - user query is usually **shorter** than entries in the corpus. Flipping the query and entries in the corpus does not make sense.

Cosine Similarity vs Dot Product



- Cosine Similarity normalizes vectors to magnitude of 1, so it can handle documents of varying length.
- Dot Product is influenced by the magnitude of the vectors, so longer documents might have larger magnitudes.

Carnegie Mellon University

What sort of embeddings will work?

- TFIDF
- Word Vectors
- BERT
- GPT

*Let's go to the river **bank***
vs.
*I need to deposit a check at the **bank***

BERT vs SBERT

SBERT [2]

- *Sentence-BERT(SBERT) is a modification of the pretrained BERT network that use siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity.*
- SBERT can be easily fine-tuned

[2] Reimers, Nils, and Iryna Gurevych. *Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks*.

Model Selection

Multi-QA Models

The following models have been trained on 215M question-answer pairs from various sources and domains, including StackExchange, Yahoo Answers, Google & Bing search queries and many more. These model perform well across many search tasks and domains.

These models were tuned to be used with dot-product:

Model	Performance Semantic Search (6 Datasets)	Queries (GPU / CPU) per sec.
multi-qa-MiniLM-L6-dot-v1	49.19	18,000 / 750
multi-qa-distilbert-dot-v1	52.51	7,000 / 350
multi-qa-mpnet-base-dot-v1	57.60	4,000 / 170

These models produce normalized vectors of length 1, which can be used with dot-product, cosine-similarity and Euclidean distance:

Model	Performance Semantic Search (6 Datasets)	Queries (GPU / CPU) per sec.
multi-qa-MiniLM-L6-cos-v1	51.83	18,000 / 750
multi-qa-distilbert-cos-v1	52.83	7,000 / 350
multi-qa-mpnet-base-cos-v1	57.46	4,000 / 170

Carnegie Mellon University

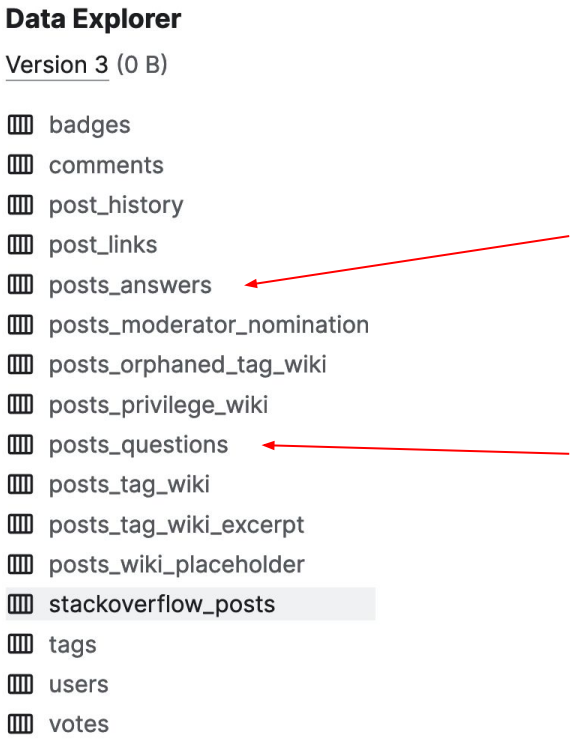
How to store user posts and their huge embeddings?

- Library that allows developers to quickly index and search for embeddings
- Think of it as the infrastructure for VectorDBs

Data Processing

To create a semantic search engine on StackOverflow, we need data that matches a post closely to user queries.

StackOverflow Archive Data Overview



The data can be fetched using Google Big Query API

Carnegie Mellon University

Data Processing

	question_id	question_title	question_body	question_score
0	53817373	How do I access HttpContext in Server-side Bla...	I need to access <code>HttpContext</code> i...	34
1	36091902	pandas: how to find the most frequent value of...	how to find the most frequent value of each...	13
2	36141186	What Version of Maven is Compatible with Java 6?	I have to work in an older project that req...	20
3	36044275	golang "go get" command showing "go: missing G...	I'm new in go lang. Trying to import a go l...	31
4	54401851	What is the difference between React Native an...	What is the difference between <a href="htt...	36

Before

* I have installed Node.js modules by 'npm install' *

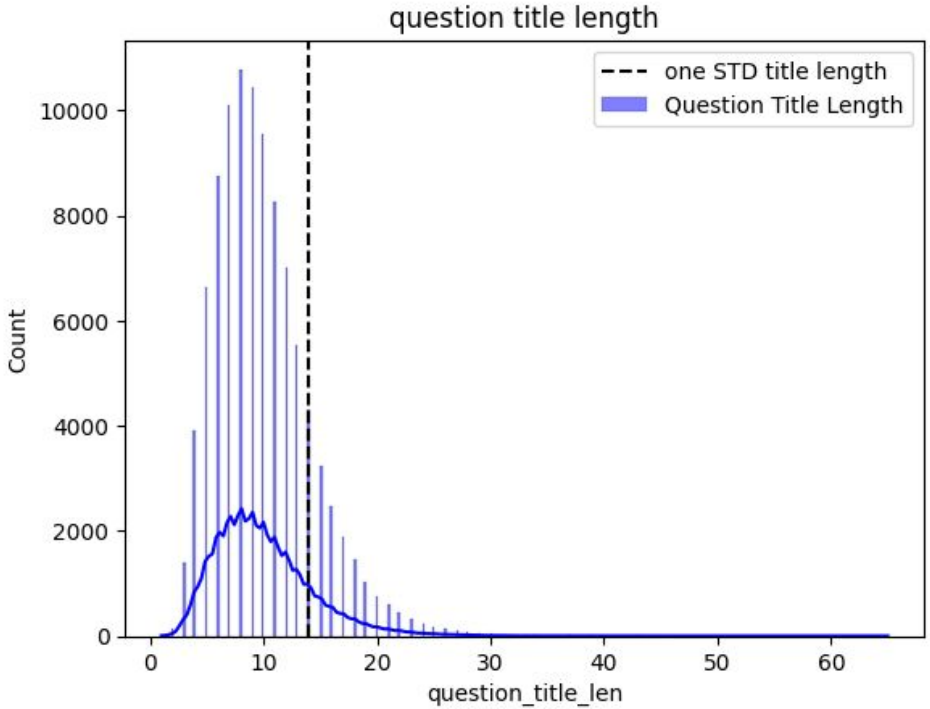
After

I have instal Node.js module by 'npm install'

- Remove HTML tags and extra whitespaces
- Decontract words (e.g. I'm -> I am, can't -> cannot)
- Lemmatized the sentences

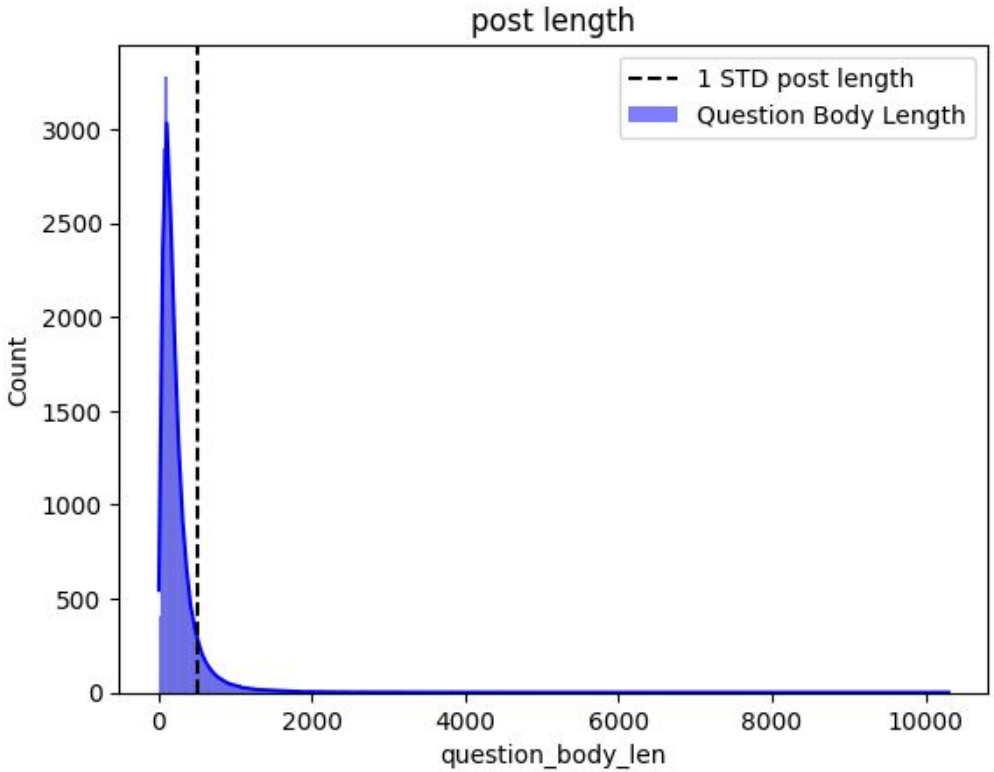
Carnegie Mellon University

Question title Length



Mean: 9.9564
STD: 4.2497

Question Post Length

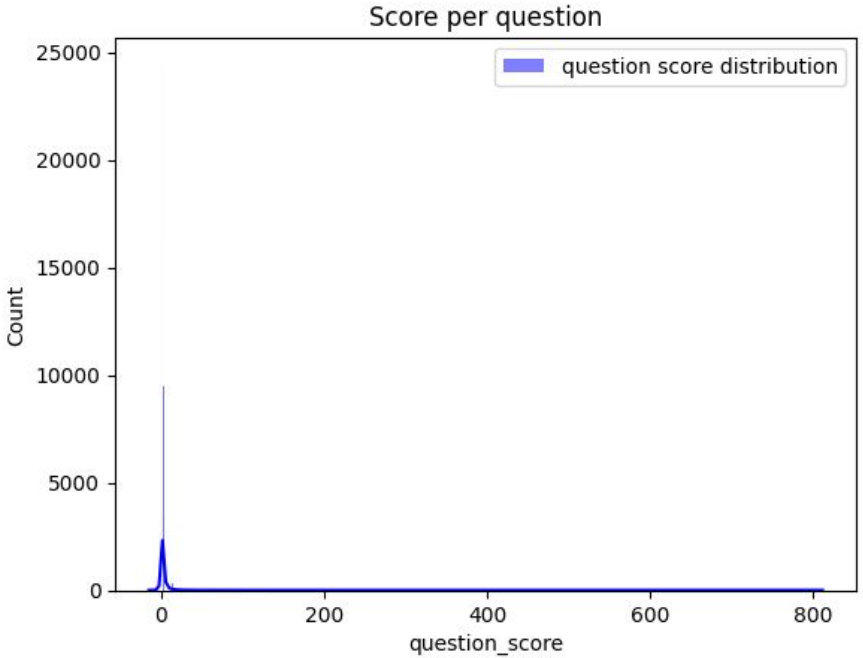


Mean: 240.3619

STD: 275.3093

Carnegie Mellon University

Question Score



Mean: 3.1619
STD: 11.0470

Carnegie Mellon University

Handling the token size limit for embedding models

The SBERT models have a token limit of 512, we need a way to handle posts longer than 512 tokens.

1. Remove all posts where question_body token size is greater than 512.
2. Summarize the question posts.
3. Create multiple embeddings for a single post, with X% overlap in content between each embedding.

Creating embeddings

```
def create_embedding(model, index_name):
    print('Creating embeddings...')
    embeddings = np.array([model.encode(text) for text in df['lemmatized_question_body']]).astype('float32')
    # Create a FAISS index
    index = faiss.IndexFlatL2(embeddings.shape[1])
    index.add(embeddings)
    faiss.write_index(index, f'{PATH_TO_DATA}{index_name}')
    return index
```

Search embeddings

```python
def fetch_post_info(df_idx):
    info = df.iloc[df_idx]
    meta_info = {
        'question_title': info['question_title'],
        'question_body': info['question_body'],
        'question_score': info['question_score'],
        'question_url': f"https://stackoverflow.com/questions/{info['question_id']}"
    }
    return meta_info


Codeium: Refactor | Explain | Generate Docstring | ✕ | Codiumate: Options | Test this function
def search(query, index, model, top_k=10):
    start_time = time.time()
    query_embedding = model.encode([query])
    search_results = index.search(query_embedding, top_k) # returns [[distance], [index]]
    end_time = time.time()
    print(f'Total search time: {end_time - start_time:.4f} seconds')

    # print(search_results)
    top_k_idx = np.unique(search_results[1]).tolist()
    top_k_posts = []
    for idx in top_k_idx:
        top_k_posts.append(fetch_post_info(idx))
    return top_k_posts
```

Carnegie Mellon University

Pretrained model results (v1 & baseline)

Query: “How to sort integers in python”

1. Sort a list from an index to another index
2. Python custom comparator to sort a specific list
3. Python list folders list by numeric order
4. TypeError: notop list' object does not support indexing
5. Sort dataframe based on first digit of a column
6. Sorting list of tuples based on results of operation (division)
7. Sorting numeric String in Spark Dataset
8. Python sorting list with negative number
9. Python default "sorted" use Merge or Quick sort? What algorithm used?
10. Python sorting numbers on last digit?

Fine-tuning the Model

Fine-tune the model

We could have easily fine-tuned the sentence-transformer model if we had data regarding **query & relevant posts** information. The problem is this data would never exist if we are building a search engine from scratch (e.g. a company's internal wiki) or the current data isn't good (StackOverflow search engine)

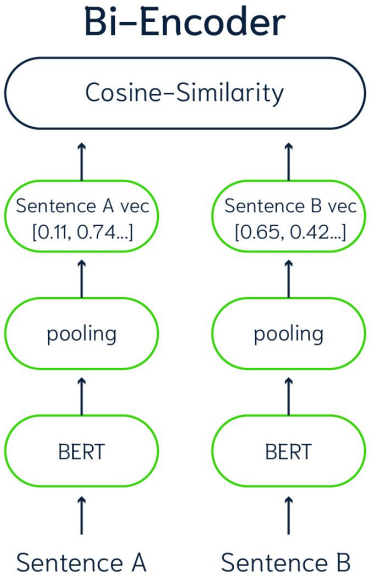
Synthetic Query Generation

- Use the question post as the knowledge base, and then generate possible user queries based on that.
- Model for generating synthetic queries: BeIR/query-gen-msmarco-t5-large-v1 [1]
- We want pairs of (query, post)

[1] Thakur, Nandan, et al. “BEIR: A Heterogenous Benchmark for Zero-Shot Evaluation of Information Retrieval Models.” *ArXiv:2104.08663 [Cs]*, 20 Oct. 2021, arxiv.org/abs/2104.08663.

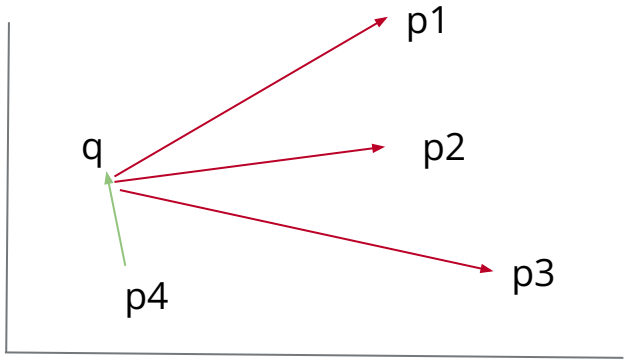
SBERT fine-tuning

- SBERT is a bi-encoder using mean pooling for encoding and cosine-similarity for retrieval



Loss Function

- MultipleNegativesRankingLoss
- Batch = of [(qi, pi), ..., (qn, pn)]
- (query, post)
- Minimize distance between (qi, pi)
- Maximize distance between (qi, pj), where i != j



Why not CosineSimilarityLoss?

Fine-Tuned model results (v2)

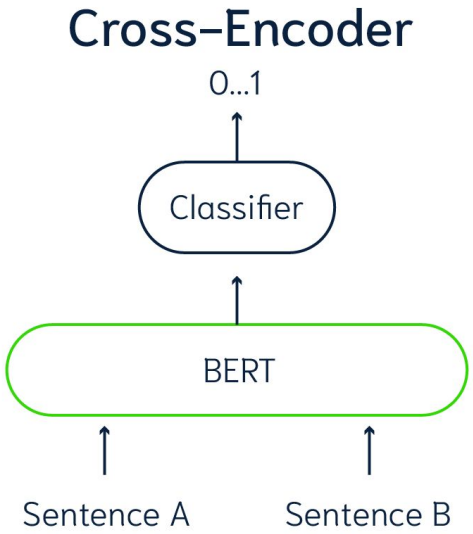
Query: “How to sort integers in python”

1. Sort a list from an index to another index
2. Python custom comparator to sort a specific list
3. Arrange list of strings that are divided into 4 parts by the different parts?
4. Linux sort: how to sort numerically but leave empty cells to the end
5. Sorting list of tuples based on results of operation (division)
6. How sorting deals with key and lambda functions in two given lists
7. Python: How to perform a secondary descending alphabetic sort within a numeric primary sort
8. How to sort a list of lists with Python?
9. Python default "sorted" use Merge or Quick sort? What algorithm used?
10. Python sorting numbers on last digit?

Cross Encoders

Using Cross-Encoders to re-rank top K results

- In V2, the model was able to retrieve relevant results for the user, but it may be ranked better.



Cross-Encoders results (v3)

Query: “How to sort integers in python”

1. Python: How to perform a secondary descending alphabetic sort within a numeric primary sort
2. Sorting list of tuples based on results of operation (division)
3. Python default "sorted" use Merge or Quick sort? What algorithm used?
4. How sorting deals with key and lambda functions in two given lists
5. Arrange list of strings that are divided into 4 parts by the different parts?
6. Sort a list from an index to another index
7. Linux sort: how to sort numerically but leave empty cells to the end
8. Python custom comparator to sort a specific list
9. Python sorting numbers on last digit?
10. How to sort a list of lists with Python?

“Process vs Thread”

Pretrained

How do node.js and libuv use the different threads?

Difference between Action as parameter and plain lambda as parameter

Threads vs cores when threads are asleep

Fine-tuned

Threads vs cores when threads are asleep

Python - serial process, multthreading, muliprocess all taking same time to run in my local

C++ will thread created in class share the same class variable?

Cross Encoders

pthread_join and pthread weird behaviour

Python threading/multiprocessing do not need Mutex?

Node.js multithreading: What are Worker threads and how does it work?

Evaluation and Future Plans

Improving the system in the long run

- Use metrics to determine whether a post matches a query.
  - “Good” and “Bad” button at bottom of every post
  - Monitor user activity by upvotes on posts
  - User clicks
  - User time spent on a post

- Score = 0.5x + 0.3y + 0.2z
- Evaluate the model on this dataset, then fine-tune it and use next time frame’s (e.g. monthly) data as evaluation again.

Evaluation methods

- Now that we have data for the top K posts given a query, we can calculate precision and recall at K.
- **Precision at K:** the number of relevant items among the top K retrieved items divided by K.
- **Recall at K:** the number of relevant items among the top K retrieved items divided by the total number of relevant items (across the entire dataset).

Future Applications

Semantic Search can be applied outside of just StackOverflow. It can be used across any website that has a search bar, including documentation pages, internal company wikis, codebase, etc.

- Embed longer content, e.g. a documentation page will be longer than 512 tokens
- With longer content, we can also embed answers along side with questions
- Support multiple media formats (images, videos)
- Try more embedding models (e.g. BERT vs GPT)
- Explore other ways of fine-tuning the model
- Explore hybrid search models

References

[1] Thakur, Nandan, et al. “BEIR: A Heterogenous Benchmark for Zero-Shot Evaluation of Information Retrieval Models.”

ArXiv:2104.08663 [Cs], 20 Oct. 2021, arxiv.org/abs/2104.08663.

[2] Reimers, Nils, and Iryna Gurevych. *Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks*.