GC-183 Deep Learning Search Engine

Abstract

Most of the search engines use algorithms like Best Match 25 [1] that perform well and return the top-ranked results, but they lack the ability to understand the semantics of the user that they are searching for. The main task of this project is to apply the various deep learning techniques to build the search engine that gives the most relevant results using the search query using the pre-trained models. Our main objective is to use deep learning to rank highly similar results at scale. This project also deals with the image data using the Image Captioning model that was trained on the Open Images V6 dataset [2]. We have successfully vectorized the text data from the 200,000+ Jeopardy! Questions dataset [3] and wrote the search engine to search given query by vectorizing the search query and return results with the least cosine or euclidean distance

Introduction

Nearest neighbor search (NNS) selects chunk of data from the database that has the smallest distance to the given query. Despite much research in this domain it is generally very expensive to find the nearest neighbor in the high dimensional euclidean or cosine space because of the curse of dimensionality [4]. We used the Image Captioning and Approximate Nearest Neighbor search (ANNs) to rank the similar results with the closest proximity using Microsoft's SPTAG. To the best of our knowledge, at the time of writing this paper, there is no such implementation or work that has been done so far that uses Microsoft's SPTAG and Image Captioning at the same time to perform the nearest neighbor search.

Research Question(s)

How can we search in a very high dimensional data for the similar data at scale?
How do we evaluate the search results?

Methodology

We used the Microsoft's Space Partition Tree And Graph (SPTAG) [5] library, which is built by Microsoft used in Bing's core search engine [6]. The proposed architecture of the model workflow is shown in Fig. 1. This library is for a large-scale vector approximate nearest neighbor search scenarios released by Microsoft Research (MSR) and Microsoft Bing. It uses the vector data to perform the nearest neighbor search.

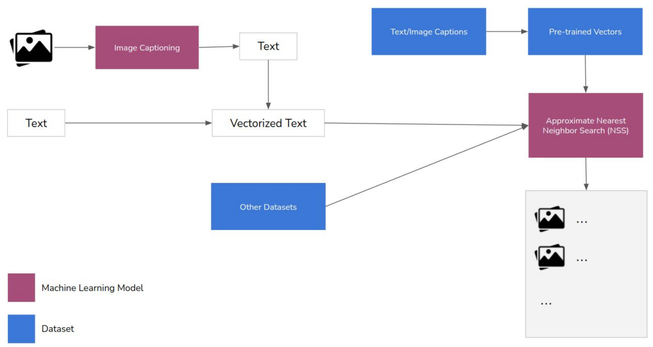

Fig. 1 High level architecture of our system

Results

For this work we used two different datasets. One dataset is text-based and the other dataset is image-based with captions. The first dataset that we used for searching text and indexing is the 200,000+ Jeopardy! Questions [3] from Kaggle, which contain the jeopardy questions and answers. For the Image Captioning model we used the Open Images V6 dataset that adds localized narratives.

We performed data preprocessing and transformation into the vectors using the Universal Sentence Encoder model and indexed the vectored data using the Microsoft's SPTAG with the Balanced k-means Tree and relative neighborhood graph (SPTAG-BKT) algorithm as it has very good search accuracy with very high-dimensional data as shown in Fig. 4. Our dimension of the dataset is 216,930 × 512. We then successfully deployed our model to the server using the Flask along with the exported pre-trained models state dictionaries. When given the search query like "Linux" the model was able to rank the results with "Operating Systems", "Windows", etc. even though the dataset doesn't contain the word "Linux" in it. The response of the results was within 0.5 seconds. The Fig. 2 and Fig. 3 shows our system workflow and result.

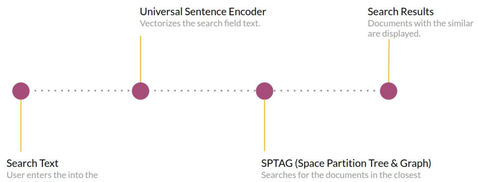

Fig. 2 Search workflow of the system

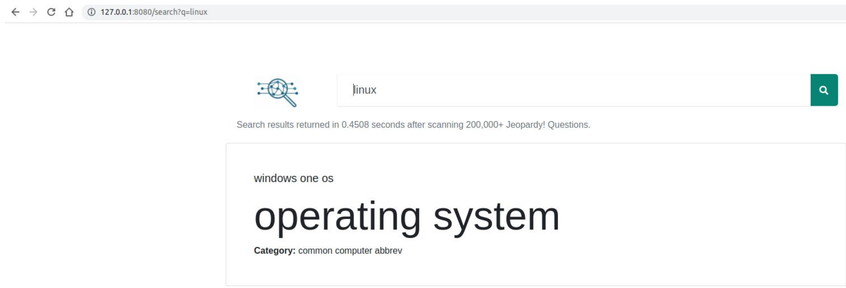

Fig. 3 Results for the search query "linux"

Fig. 4 Vectorized data

Conclusions

We successfully preprocessed, trained, and deployed the model onto the server. The indexer was built using the Balanced k-means Tree and relative neighborhood graph (SPTAG- BKT) algorithm. Then, we trained the image captioning model to return captions (e.g., text) from the image and then feed it to the vectorizer that generates the word embedding using Universal Sentence Encoder as shown in the architecture in Fig. 2. For the future work, we can try indexing the text corpus using the kd-tree and relative neighborhood graph (SPTAG-KDT) and compare its performance

Flask TensorFlow PyTorch

Acknowledgments

We would like to thank Dr. Md Abdullah Al Hafiz Khan, whose expertise was invaluable in finishing this project. His insightful feedback pushed us to sharpen our thinking and brought us work to a higher level.

We would also like to thank Department of College of Computing and Software Engineering to provide us this opportunity.

Contact Information

Malik Naik Mohammed
Email: mmoham25@students.kennesaw.edu
Website: http://maliknaik.me/
LinkedIn: https://www.linkedin.com/in/maliknaik/

Yves Junior Kwame Kamgaing - ykwameka@students.kennesaw.edu

Madhusudhan Rao Atmakuri - matmakur@students.kennesaw.edu

Sai Charan Gandi - sgandi@students.kennesaw.edu

References

[1] Wikipedia, "Okapi BM25 — Wikipedia, the free encyclopedia," https://en.wikipedia.org/wiki/Okapi BM25, 2022, [Online; accessed 15-April-2022].
[2] "Open images dataset v6 + extensions." [Online]. Available: https://storage.googleapis.com/openimages/web/index.html
[3] B. Tunguz, "Towards efficient and intelligent internet of things search engine," Kaggle.com, vol. 9, pp. 15 778–15 795, 2021.
[4] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, ser. STOC '98. New York, NY, USA: Association for Computing Machinery, 1998, p. 604–613. [Online]. Available: https://doi.org/10.1145/276698.276876
[5] Q. Chen, B. Zhao, H. Wang, M. Li, C. Liu, Z. Li, M. Yang, and J. Wang, "Spann: Highly-efficient billion-scale approximate nearest neighbor search," in 35th Conference on Neural Information Processing Systems (NeurIPS 2021), 2021.
[6] C. Waldburger, "'how tall is the tower in paris?' how bing knows its about the eiffel tower," Dec 2019. [Online]. Available: https://blogs.microsoft.com/ai/bing-vector-search

KENNESAW STATE UNIVERSITY
COLLEGE OF COMPUTING AND SOFTWARE ENGINEERING

Author(s): Malik Naik Mohammed, Yves Jr. Kwame Kamgaing, Madhu Rao, Charan Gandi
Advisors(s): Dr. Md Abdullah Al Hafiz Khan, mkhan74@kennesaw.edu