Journal of Innovations in Engineering Education
JIEE 2021, Vol. 4, Issue 1, pp 87-89.

https://doi.org/10.3126/jiee.v4i1.35390

THAPATHALI CAMPUS
Institute of Engineering
Tribhuvan University

Large-scale image search with text for information retrieval

Janardan Bhatta[a,*], Shekhar Koirala[b]

[a]*Institute of Engineering, Thapathali Campus, Tribhuvan University, Kathmandu, Nepal*
[b]*Identv LLC*

ARTICLE INFO

Article history:

Received 01 Jan 2021
Received in revised form
08 Feb 2021
Accepted 14 Feb 2021

Keywords:

Computer vision
Attention mechanism
Contrastive loss function
Natural language processing
Information retrieval systems

Abstract

Searching images in a large database is a major requirement in Information Retrieval Systems. Expecting image search results based on a text query is a challenging task. In this paper, we leverage the power of Computer Vision and Natural Language Processing in Distributed Machines to lower the latency of search results. Image pixel features are computed based on the contrastive loss function for image search. Text features are computed based on the Attention Mechanism for text search. These features are aligned together preserving the information in each text and image feature. Previously, the approach was tested only in multilingual models. However, we have tested it in the image-text dataset and it enabled us to search in any form of text or images with high accuracy.

1. Introduction

There are various approaches [1] [2] for the information retrieval process which requires the query to be text and the output to be Image. Here, we are dealing with two different types of data which are generally processed differently. The one approach [3] is to store all of the data including output images and probable text in a pre-defined schema forecasting all the possible use cases. But, as the data keeps on growing, it becomes hard to maintain it, and the throughput of the system also decreases which is not desired on a production scale. Here, An architecture is proposed that works in a distributed system and optimizes by state-of-the-art feature extraction methods, and still tackles the problem of the multivariate data source of text and image in a query. The system first calculates the image features. Image features are extracted based on the Generalized Image Detection model [4] which detects various landmarks inside an image frame. The detected landmark will be used to generate features based on an encoder model. In the second step, text features are calculated from the label of the image dataset based on a language model. At this point, Only the power of data type dependent models have been utilized but can be used only separately. To combine these features, a multi-domain model was created in order to learn a mapping from the text to the image feature using Procrustes alignment.

*Corresponding author:
bhattajanardan@gmail.com (J. Bhatta)

2. Architecture overview

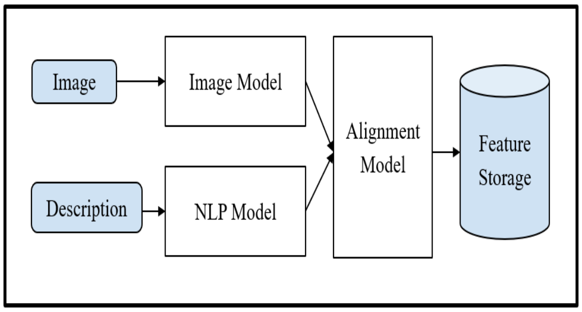


Figure 1: Model architecture flow

In this paper, images with its description were fed into two models. The image model consisted of a Detector model and the embedding model. An image forwarded to the image model outputs various localized detections and embedding is calculated through the embedding model. Similarly, The description was forwarded to the NLP model which outputs text embedding. Both embeddings were aligned together based on Muse Architecture [5]. The aligned embedding was then stored in the distributed vector storage [6]. This enabled us to search the expected image, to be searched either by images or by text similar to what we had in the description. The model inference was done as a distance query with Anisotropic Vector Quantization.

2.1. Image features

The simCLR [7] architecture is picked for calculating the features of images. The architecture depended on two random image augmentation. The backbone of the model was Resnet50 [8]. A small fully connected linear neural network was used to project the backbone model's embedding into another space. In the end, the contrastive loss was calculated. The loss value was decreased when the images were similar.

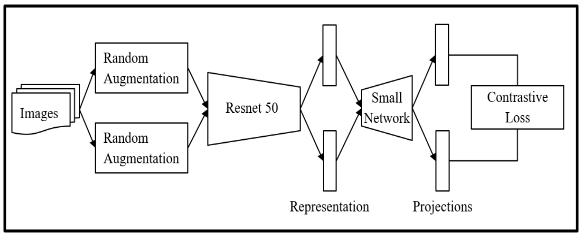


Figure 2: Image feature extraction process

It is important to note that the contrastive loss function proceeds in a batch of images. The value of the loss is less for the same positive pair of images with augmentation. Whereas, for a negative pair of images, the loss would be more. Definition of the loss for pair of positive examples (i) and (j) defined as [8]:

$$l_{1,2} = -log \frac{\exp[sim(z_i, z_j)/\tau]}{\sum_{n=1}^{2n} \mathbb{1}_{[k\neq 1]} \exp[sim(z_i, z_j)/\tau]} \quad (1)$$

Where $1[k6 = i] \in \{0, 1\}$ is an indicator function evaluating to 1 iff $k \neq i$ and τ denotes a temperature parameter. An appropriate temperature is needed for a model to learn from hard negatives. In this paper, the value of τ was taken as 0.01 based on the analysis done in the original paper [7]

In our implementation, the base model of frozen Resnet50 was trained on the Imagenet21k dataset, and gradients were calculated in a single layer only. This also yielded higher accuracy. The learning rate for the model was calculated based on the LRfinder algorithm [9] implemented based on the *PyTorch-lightning* library. Batch size of 256 and epoch of 50 performed well for the embedding extraction process. The small network that is used after the image representation vectors calculated, consisted of two linear layers combined with a ReLU activation unit. The size of both layers is based on the size of the embedding which is 512. The output embedding was then stored in distributed vector storage.

2.2. Text features

Text features representation was calculated on text sequence based on self-attention mechanism. Attention function consists of the query, key-value pair, and the output, where each of the entities is vectors [10].

$$\text{Attention(Q,K,V)} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (2)$$

The model architecture is the same as GPT-2 architecture [10]. Just like Image embeddings, Text embeddings are also stored in vector storage and ready to send both embeddings to the alignment model

2.3. Alignment model

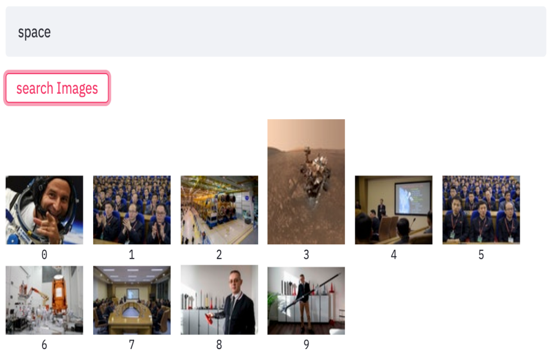
The features from both the image and text model were differently aligned. In order to use both features in a single database and query, the vectors need to be aligned in a single vector space. The alignment model [5] was used in the multilingual language model but it did not perform well as there are many rare words and can be used in different contexts. However, in our case of image-text, a dataset has been sampled and a balance labeled dataset is used in order to achieve a higher alignment model. The task of the alignment model is to learn mapping W without any cross-domain, image-text, supervision [5].

$$\begin{aligned} W^* &= \text{argmin}_{w \in O_d(r)} \parallel WX - Y \parallel_F \\ &= UV^T \\ \text{With} & \\ U \sum V^T &= SVD(YX^T) \end{aligned} \quad (3)$$

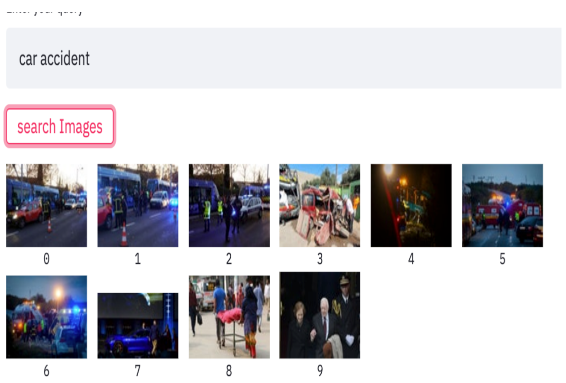
The domain-adversarial approach is used for learning W. let's take $X = \{x_1, x_2, ..., x_n\}$ and $Y = \{y_1, y_2, ..., y_m\}$ two $_n$ and $_m$ word embeddings coming from image and text respectively. A discriminator model will train in order to differentiate between $WX = \{Wx_1, ..., Wx_n\}$ and Y. On contrast,W, mapping matrix, is trained in order to fail the discriminator model to differentiate the two vectors. The discriminator and the W are trained successively with stochastic gradient updates in each input image step. A synthetic global dictionary is built with CSLS (Cross-domain similarity local scaling). Since Images don't have any vocabulary, a custom index is assigned to label the output of images. The mean cosine of embedding pair based on the dictionary is chosen as validation metrics for the alignment model. The mapping W is further optimized by Procrustes optimization. Once the text and image embeddings are aligned, the alignment matrix is saved for the inference cycle.

3. Performance

For each inference cycle, embeddings are generated based on user queries. Those embedding are then aligned using the alignment model and searched in the vector space. A Distributed system vector space is used in order to store vectors in millions scale. As the Distributed system stores vectors in a clustered fashion. The inference querying cycle is heavily optimized using the center of cluster to search at first and then using the nearest units to get all the output. Some results of the system are shown in Figure 3.



(a) Keyword: Space



(a) Keyword: Car accident

Figure 3: Search example

In order to perform these operations, nearly 1 Million images were scraped from the web and ran through the model. The performance of the model depends of topk of the result fetched. The performance-based on topk is shown in Table 1.

Table 1: Result table based on the time taken and query size based on accuracy

topK	Time taken (msec)	Accuracy (%)
1	0.02	99
10	0.08	97
100	0.2	91
1000	0.3	76

4. Conclusion

The accuracy of the system increases with the combination of image features as well as the text features. The usability of this particular Information retrieval is more as the inference speed is below second. The use of a distributed database system also ensures less latency even though topk of the query is increased.

5. Acknowledgement

The research is supported by *IOE, Thapathali Campus*, for technical as well as financial support for all phases of the research. The data and code resources are the proprieties of *Identv LLC*.

References

[1] Surdeanu M, McClosky D, Smith M, et al. Customizing an Information Extraction System to a New Domain[J/OL]. Proceedings of the ACL 2011 Workshop on Relational Models of Semantics, RELMS '11, 2011(Relms): 2-10. http://dl.acm.org/citation.cfm?id=2021153.2021155.

[2] lemmond, t. , hanley, w. , guensche, j. , perry, n. , nitao, j. , kidwell, p. , boakye, k. , glaser, r. , & prenger R. Information extraction system[J]. 2014.

[3] Muslea I. Extraction Patterns for Information Extraction Tasks: A Survey[J/OL]. The AAAI99 Workshop on Machine Learning for Information Extraction, 1999: 1-6. http://scholar.google.com/scholar?hl=en{Ź}btnG=Search{Ź}q=intitle:Extraction+Patterns+for+Information+Extraction+Tasks:+A+Survey{}}0.

[4] Song L, Li Y, Jiang Z, et al. Fine-grained dynamic head for object detection[J]. arXiv, 2020.

[5] Conneau, A. , Lample, G. , Ranzato, M. , Denoyer, L. , & Jégou H. Word Translation Without Parallel Data[J]. 2018.

[6] Guo, R. , Geng, Q. , Simcha, D. , Chern, F. , Sun, P. , Lindgren, E. , & Kumar S. Accelerating Large-Scale Inference with Anisotropic Vector Quantization[J]. 2020.

[7] T. Chen, M. Norouzi S K, Hinton G. A Simple Framework for Contrastive Learning of Visual Representations[J]. 2020.

[8] he, k. , zhang, x. , ren, s. , & sun J. Deep Residual Learning for Image Recognition.[J]. 2016.

[9] Smith L. Cyclical Learning Rates for Training Neural Networks[J]. 2017: 464-472.

[10] Radford, A. , Wu, J. , Child, R. , Luan, D. , Amodei, D. , & Sutskever I. Language Models are Unsupervised Multitask Learners[J]. 2019.