Text to Image Search using CLIP - Sentence Transformers

Ashwini Kumar, Thevananthan Thevarasa and Ismat Sifat Yousuf

Abstract—Text-to-image search is a challenging task in the field of computer vision that requires an understanding of the semantic relationship between text and images. Recent advances in natural language processing (NLP) and computer vision have enabled the development of powerful models for solving this task. In this work, we propose a novel approach for text-to-image search using a combination of CLIP, a language-model-based image retrieval model, and sentence transformers, a powerful sentence encoding model. Our proposed method enables retrieving images that semantically match the given input text by encoding the text and image representations into a shared embedding space. We evaluated our approach on a large-scale image dataset, and our experimental results show that the proposed method outperforms the state-of-the-art methods in terms of retrieval accuracy. Our work provides a promising direction toward developing more accurate and robust text-to-image search systems for various applications, including e-commerce, digital marketing, and image recommendation systems.

Index Terms—Text to Image Search, CLIP, Sentence Transformers, Machine Learning.

1 Introduction

TEXT-to-image retrieval is a burgeoning research area that aims to retrieve images based on textual queries. To improve the quality of image retrieval, recent studies have employed sentence transformers in text-to-image retrieval tasks. These neural models embed text into high-dimensional vectors using a pre-trained language model, which allows similarity scores to be computed between textual and visual embeddings and enables image retrieval based on textual queries. These embeddings capture the semantic meaning of the input text and have shown significant improvements over traditional text-to-image retrieval models. Sentence transformer models can also be fine-tuned for specific tasks, such as image retrieval, to further improve performance.

In recent studies, sentence transformers have been combined with other deep learning models, such as Convolutional Neural Networks (CNNs), to achieve state-of-the-art performance in text-to-image retrieval tasks. For example, the CLIP model combines a sentence transformer with a vision transformer, resulting in a powerful text-to-image retrieval model that can retrieve images based on complex textual queries. CLIP has shown promising results in a variety of text-to-image retrieval tasks, including natural language image retrieval, image captioning, and zero-shot image classification. The model has also outperformed previous state-of-the-art methods in text-to-image retrieval benchmarks, such as the COCO dataset

- *Ashwini Kumar, Thevananthan Thevarasa and Iismasmat Sifat Yousuf are the students of Masters of Artificial Intelligence at the Memorial University of Newfoundland.*
- *E-mail: ashwinik@mun.ca , thevananthat@mun.ca and isyousuf@mun.ca*

2 Related Work

The utilization of text feedback in image search has potential benefits in real-world applications, including e-commerce and internet search. This task aims to retrieve images that resemble the input image and simultaneously modify certain features based on the provided text feedback. This is a complex task that requires a comprehensive understanding of both text and image. In this study, we propose a novel framework called Visiolinguistic Attention Learning (VAL) to tackle this challenge. Specifically, we present a composite transformer that can be incorporated into a CNN to selectively maintain and modify the visual characteristics based on language semantics. By inserting multiple composite transformers at different levels, VAL is encouraged to capture multi-granular visiolinguistic information, leading to an expressive representation for efficient image retrieval [1].

This paper presents a hybrid image retrieval system for the World Wide Web that leverages both text and image content features. The system uses a text-based image meta-search engine to retrieve a large initial set of images based on the text information on the image host pages. This approach provides a high recall rate and is both fast and cost-effective. Next, an image content-based ordering is performed on the initial image set, clustering images into different folders based on their content features. The system also allows re-ranking of the images based on user feedback. This approach enables the practical use of both text and image content for image retrieval over the Internet. The experimental results confirm the system's efficiency [2].

VisualSparta: An Embarrassingly Simple Approach to Large-scale Text-to-Image Search with Weighted Bag-of-words in cross-modal information retrieval, text-to-image retrieval is a crucial task that involves retrieving relevant images from a large and unlabeled dataset based on textual queries. In this paper, we present VisualSparta, a novel

model based on Visual-text Sparse Transformer Matching, which exhibits remarkable improvements in both accuracy and efficiency. VisualSparta has been demonstrated to outperform previous state-of-the-art scalable approaches in MSCOCO and Flickr30K. Furthermore, we illustrate that VisualSparta provides substantial retrieval speed advantages. For instance, when using CPU, VisualSparta can achieve 391X speedup compared to CPU vector search and 5.4X speedup compared to vector search with GPU acceleration for a 1 million image index. Our experiments also indicate that this speed advantage becomes more significant for larger datasets since VisualSparta can be efficiently implemented as an inverted index. To the best of our knowledge, VisualSparta is the first transformer-based text-to-image retrieval model that can realize real-time searching for large-scale datasets while offering significant accuracy improvements over previous state-of-the-art methods [3].

"What can visual content analysis do for text-based image search?" discusses text-based meta word search engines such as Google, Yahoo!, and Microsoft Live image search are widely used for image search. These search engines rely on text queries to rank image search results, neglecting the rich visual information in the images themselves. However, recent advancements in these search engines, particularly in Microsoft Live image search, have introduced new features that analyze the visual content of images. In this paper, we provide an overview of these features, discuss their design principles, and propose the development of new content analysis-based features for text-based image search engines [5].

iLike: Bridging the Semantic Gap in Vertical Image Search by Integrating Text and Visual Features presents with the increasing availability of multimedia content on the Internet and the emergence of Web 2.0, there is a growing need for efficient and accurate retrieval of images that meet user's requirements. While content-based image retrieval (CBIR) has been extensively researched in the academic community, text-based search is more widely used in the industry. However, both approaches have their own limitations and disadvantages, and current web image search engines still face challenges. In this study, we introduce iLike, a vertical image search engine that combines both textual and visual features to enhance retrieval performance. We bridge the semantic gap by capturing the meaning of each text term in the visual feature space and weighting visual features based on their relevance to query terms. Additionally, we address the user intention gap by inferring the "visual meanings" of textual queries. We also provide a visual thesaurus generated from the statistical similarity between visual space representations of textual terms. Our experimental results demonstrate that our approach improves both precision and recall compared to content-based or text-based image retrieval techniques. Most notably, search results from iLike are more consistent with users' perceptions of query terms [6].

Zero-Shot Text-to-Image Generation describes a simple approach for text-to-image generation based on a transformer that auto-regressively models the text and image tokens as a single stream of data. With sufficient data and scale, the approach is competitive with previous domain-specific models when evaluated in a zero-shot fashion [7].

Controllable Text-to-Image Generation proposes a novel controllable text-to-image generative adversarial network (GAN) that discusses the conventional approach for text-to-image generation, which has concentrated on enhancing modelling assumptions to train on a predefined dataset. These assumptions might include intricate architectures, auxiliary losses, or additional information like object part labels or segmentation masks provided during training. In this work, we introduce a straightforward method for this task based on a transformer that models the text and image tokens as a single stream of data in an autoregressive manner. When evaluated in a zero-shot manner, our approach is as effective as previous domain-specific models, given sufficient data and scale [8].

3 ARCHITECTURE

CLIP (Contrastive Language-Image Pre-Training) is a machine learning model introduced by OpenAI in 2021 designed to understand natural language and visual information and perform a wide range of tasks requiring this understanding. The model comprises a text and image encoder trained using a contrastive learning framework. The text encoder, a transformer, generates a fixed-length vector representation of the text, while the image encoder, a convolutional neural network (CNN), generates a fixed-length vector representation of the image. The CLIP model maximizes a similarity metric between matching pairs of text and image representations while minimizing the similarity metric between non-matching pairs. Once trained, the CLIP model can be used for a variety of tasks, such as image classification and retrieval.

The Contrastive Pre-training: Assume we have a batch of N images paired with their respective descriptions e.g. (image1, text1), (image2, text2), (imageN, textN). Contrastive Pre-training aims to jointly train an Image and a Text Encoder that produce image embeddings [I1, I2 ... IN] and text embeddings [T1, T2 ... TN], in a way that: • The cosine similarities of the correct (image-text) embedding pairs (I1,T1), (I2,T2) (where i=j) are maximized. • In a contrastive fashion, the cosine similarities of dissimilar pairs (I1,T2), (I1,T3)... (Ii,Tj) (where i =/ j) are minimized figure 1 1.

Let's see what happens step-by-step:

1. The model receives a batch of N ¡image-text¿ pairs.
2. The Text Encoder is a standard Transformer model with GPT2-style modifications. The Image Encoder can be either a ResNet or a Vision Transformer.
3. The Image Encoder computes an image vector for every image in the batch. The first image corresponds to the I1 vector, the second to I2, and so on. Each vector is of size de, where de is the size of the latent dimension. Hence, the output of this step is N X de matrix.
4. Similarly, the textual descriptions are squashed into text embeddings [T1, T2 ... TN], producing an N X de matrix.
5. Finally, we multiply those matrices and calculate the pairwise cosine similarities between every image and text description. This produces an N X N matrix, shown in Figure 1.
6. The goal is to maximize the cosine similarity along the diagonal — these are the correct ¡image-text¿ pairs. In a

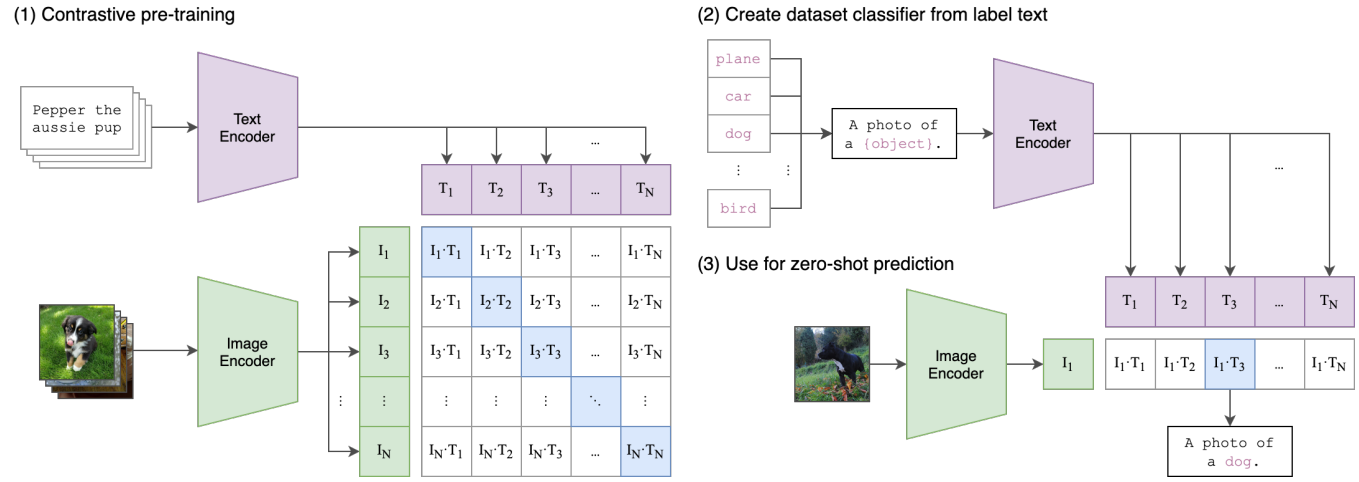

Fig. 1. An overview of the Contrastive Pre-Training.

contrastive fashion, off-diagonal elements should have their similarities minimized (e.g I1 image is described by T1 and not by T2, T2, T3 etc).
A few extra remarks:
• The model uses the symmetric cross-entropy loss as its optimization objective. This type of loss minimizes both the image-to-text direction as well as the text-to-image direction (remember, our contrastive loss matrix keeps both the (I1, T2) and (I2, T1) cosine similarities).
• Contrastive Pre-training is not entirely new. It was introduced in previous models and was adapted by CLIP.
Sentence Transformer is a deep learning model used to generate embeddings for text data. These embeddings are vector representations of sentences or documents that can be used in various natural languages processing tasks like sentiment analysis, text classification, and text similarity. Sentence Transformer extends the Transformer architecture by incorporating additional training objectives, such as Siamese network training and Contrastive loss, to learn better sentence embeddings. The model has outperformed traditional embedding methods like Word2Vec and GloVe in various NLP tasks, particularly those involving sentence-level semantics. Sentence Transformer has been used in various applications like text classification, semantic search, and question-answering [9] [16].

3.1 Model Overview

The following table (figure 2) provides an overview of selected models. They have been extensively evaluated for their quality to embedded sentences (Performance Sentence Embeddings) and to embedded search queries and paragraphs (Performance Semantic Search).
The all-* models were trained on all available training data (more than 1 billion training pairs) and are designed as general-purpose models. The all-mpnet-base-v2 model provides the best quality, while all-MiniLM-L6-v2 is 5 times faster and still offers good quality.

All models

Model Name	Performance Sentence Embeddings (14 Datasets)	Performance Semantic Search (6 Datasets)	Avg. Performance	Speed	Model Size
all-mpnet-base-v2	69.57	57.02	63.30	2800	420 MB
multi-qa-mpnet-base-dot-v1	66.76	57.60	62.18	2800	420 MB
all-distilroberta-v1	68.73	50.94	59.84	4000	290 MB
all-MiniLM-L12-v2	68.70	50.82	59.76	7500	120 MB
multi-qa-distilbert-cos-v1	65.98	52.83	59.41	4000	250 MB
all-MiniLM-L6-v2	68.06	49.54	58.80	14200	80 MB
multi-qa-MiniLM-L6-cos-v1	64.33	51.83	58.08	14200	80 MB
paraphrase-multilingual-mpnet-base-v2	65.83	41.68	53.75	2500	970 MB
paraphrase-albert-small-v2	64.46	40.04	52.25	5000	43 MB
paraphrase-multilingual-MiniLM-L12-v2	64.25	39.19	51.72	7500	420 MB
paraphrase-MiniLM-L3-v2	62.29	39.19	50.74	19000	61 MB
distiluse-base-multilingual-cased-v1	61.30	29.87	45.59	4000	480 MB
distiluse-base-multilingual-cased-v2	60.18	27.35	43.77	4000	480 MB

Fig. 2. Overview of Performance of selected models

3.2 Image and Text Models

The following models can embed images and text into a joint vector space (figure 3).
The following models are available with their respective Top 1 accuracy on the zero-shot ImageNet validation dataset [13].

Model	Top 1 Performance
clip-ViT-B-32	63.3
clip-ViT-B-16	68.1
clip-ViT-L-14	75.4

Fig. 3. Top 1 accuracy on the Zero-Shot ImageNet Validation Dataset

3.3 Pre-trained Encoders

Bi-encoders and cross-encoders are two different types of neural architectures used in natural language processing (NLP) tasks, such as sentence classification, question answering, and text similarity.

A bi-encoder architecture consists of two encoder networks, one for the input text and one for the output text. The input and output encoders are identical and encode their respective inputs into high-dimensional vector representations. The similarity between two input sentences can then be computed by comparing their respective vector representations [10].

On the other hand, a cross-encoder architecture consists of a single encoder network that takes both input sentences as input and generates a single vector representation for both sentences. This single vector representation is then used to compute the similarity between the two input sentences.

The main difference between the two architectures is that bi-encoders are designed to encode individual sentences separately, whereas cross-encoders consider both input sentences together to generate a joint representation. As a result, cross-encoders can capture more complex relationships between input sentences, but they are generally more computationally expensive than bi-encoders [12].

Both bi-encoders and cross-encoders have been used in a wide range of NLP tasks, and their performance varies depending on the specific task and dataset. Some popular examples of bi-encoders include the Siamese architecture and the Universal Sentence Encoder (USE), while popular examples of cross-encoders include the BERT (Bidirectional Encoder Representations from Transformers) and RoBERTa

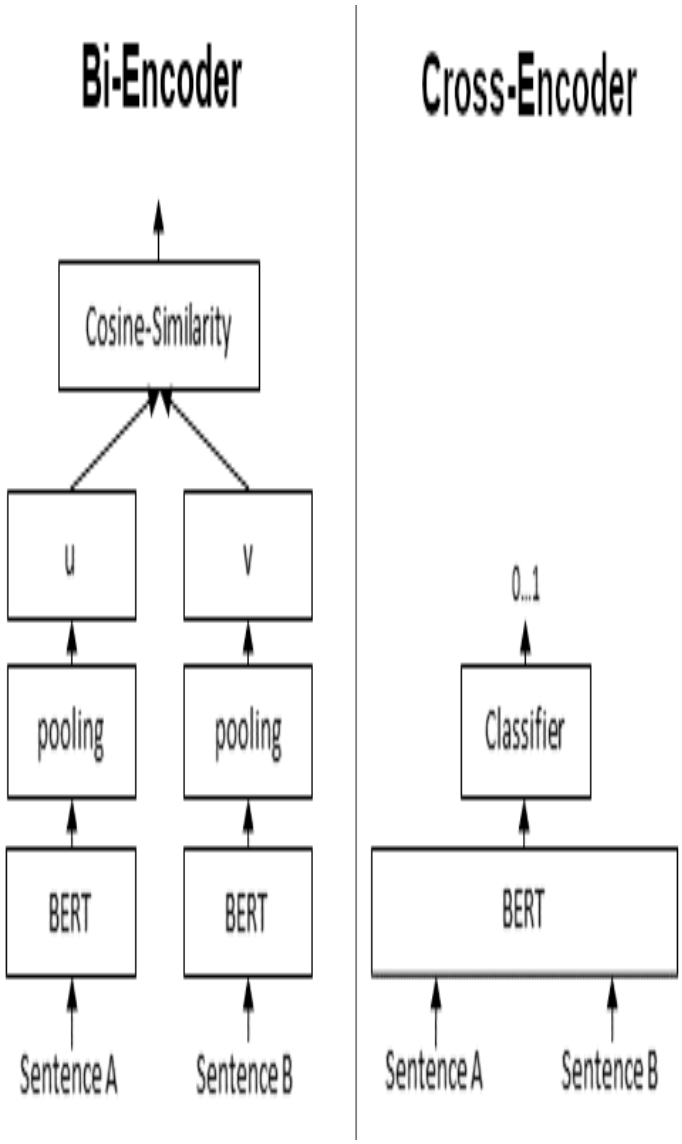

Fig. 4. Bi-Encoder vs. Cross-Encoder

models [11].

3.4 Semantic Search

Semantic search is a type of search technique that aims to understand the meaning and context behind a user's query, rather than simply matching keywords. This is achieved through the use of natural language processing (NLP) techniques, such as machine learning algorithms and deep neural networks, to analyze and interpret the query in the same way that a human would.

Semantic search aims to provide more accurate and relevant results by considering the user's intent and the context of the query. For example, if a user searches for "best Italian restaurants", a semantic search engine would not only look for web pages that contain those exact keywords but also understand that the user is looking for recommendations on Italian restaurants and provide results that match that intent.

Semantic search seeks to improve search accuracy by understanding the content of the search query. In contrast to

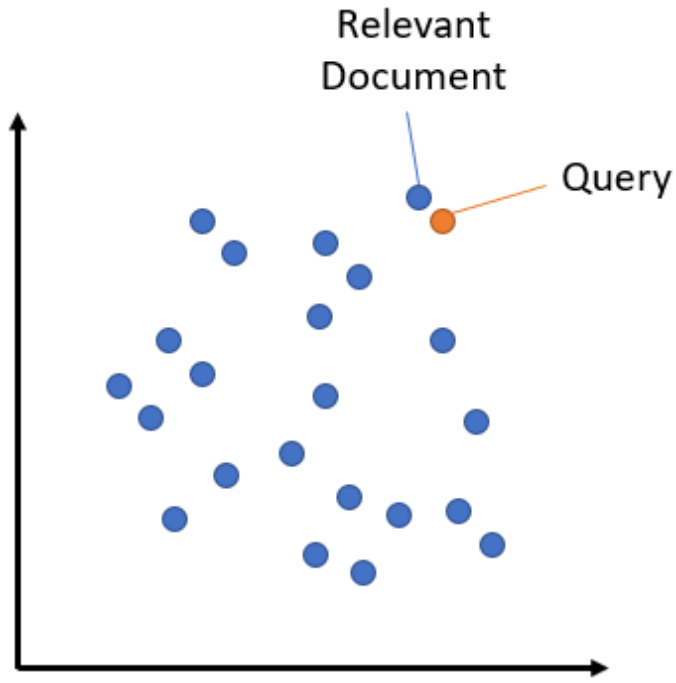

Fig. 5. Query vs Relevant Document in Semantic Search

traditional search engines which only find documents based on lexical matches, semantic search can also find synonyms. The idea behind semantic search is to embed all entries in your corpus, whether they be sentences, paragraphs, or documents, into a vector space. The query is embedded into the same vector space at search time and the closest embeddings from your corpus are found. These entries should have a high semantic overlap with the query

3.4.1 Asymmetric vs Symmetric Semantic Search

Asymmetric and symmetric semantic search are two different semantic search approaches.

In asymmetric semantic search, the query and the document representations are different. The query representation is typically a vector generated using a deep learning model, such as a transformer-based model like BERT, while the document representation is generated using traditional information retrievals techniques, such as TF-IDF or BM25. The goal of asymmetric semantic search is to retrieve documents that are most relevant to the query, based on their similarity to the query representation. For asymmetric semantic search, you usually have a short query (like a question or some keywords) and you want to find a longer paragraph answering the query. An example would be a query like "What is Python" and you want to find the paragraph "Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy ...". For asymmetric tasks, flipping the query and the entries in your corpus usually does not make sense.

On the other hand, in symmetric semantic search, both the query and the document representations are generated using the same deep learning model. This means that the same neural network is used to encode both the query and the document. The goal of symmetric semantic search is to retrieve documents that are most similar to the query, based on their similarity in the embedding space generated by the neural network. For symmetric semantic search, your query and the entries in your corpus are of about the same length and have the same amount of content. An example would be searching for similar questions: Your query could for example be, "How to learn Python online?", and you want to find an entry like, "How to learn Python on the web?". You could flip the query and the entries in your corpus for symmetric tasks.

3.4.2 Retrieve and Re-Rank

In the context of information retrieval, "retrieve and re-rank" typically refers to a two-step process for improving the ranking of search results.

In the first step, the search engine retrieves a set of candidate documents that are likely to be relevant to the user's query. This can be done using a variety of techniques, such as keyword matching, semantic search, or machine learning models.

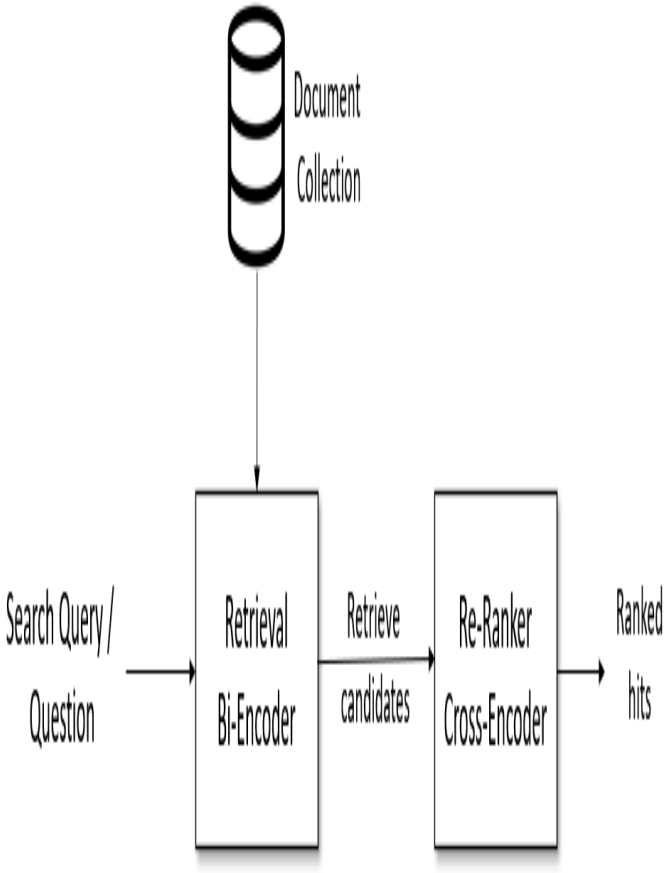

Fig. 6. Retrieve and Re-Rank Pipeline

The retrieved documents are re-ranked in the second step based on their relevance to the user's query. This is typically done by assigning a relevance score to each document based on various factors, such as the frequency of the query terms in the document, the similarity of the document to other relevant documents, or user feedback such as clicks or ratings. The documents are then sorted in descending order of their relevance score so that the most relevant documents appear at the top of the search results.

The re-ranking step is important because the initial set of retrieved documents may not be perfectly aligned with the user's query. Re-ranking allows the search engine to adjust the ranking of the documents based on additional information, such as the context of the query, the user's location, or other user-specific factors.

3.5 Training Overview

Training sentence transformers is a multistep process that involves several crucial stages. Initially, a vast corpus of text data is collected for training the model, which can be domain-specific, such as medical or legal text, or general data like Wikipedia articles. The collected data is then preprocessed by cleaning and tokenizing the text and converting it into a format that can be utilized by the model. Text normalization techniques such as removing stop words and stemming may also be applied. Next, the pre-training stage commences, where the model is pre-trained on the large corpus of text data through self-supervised learning. This involves training the model to predict missing words in a sentence or the next sentence in a sequence, enabling it to learn general language representations. The pre-trained model is then fine-tuned on a specific task, such as text classification, semantic similarity, or text-to-image retrieval, using a smaller, task-specific dataset to adjust its pre-trained language representations to the specific task. Subsequently, the model's performance is evaluated on a validation set to assess its accuracy and effectiveness for the specific task. Further, hyperparameters of the model, such as learning rate, batch size, and the number of epochs, are adjusted to optimize its performance on the validation set. Finally, the trained model is deployed for use in real-world applications like search engines or chatbots, where it can generate embeddings for new text inputs and perform a semantic search or other NLP tasks [17].

In order to achieve optimal performance for a given task, it is essential to fine-tune sentence/text embeddings specifically for that task. SentenceTransformers facilitates this process by offering a variety of building blocks that can be combined to create custom models for individual tasks. However, since each task is unique, there is no one-size-fits-all training strategy. The optimal training strategy will depend on the available data and the target task.

3.6 Network Architecture

The architecture for sentence/text embeddings can vary depending on the specific approach used. However, many recent approaches use deep neural networks such as convolutional neural networks (CNNs) or transformer-based architectures such as BERT.

For sentence/text embeddings, we want to map a variable-length input text to a fixed-sized dense vector. The most basic network architecture we can use is the network architecture shown in Figure 7.

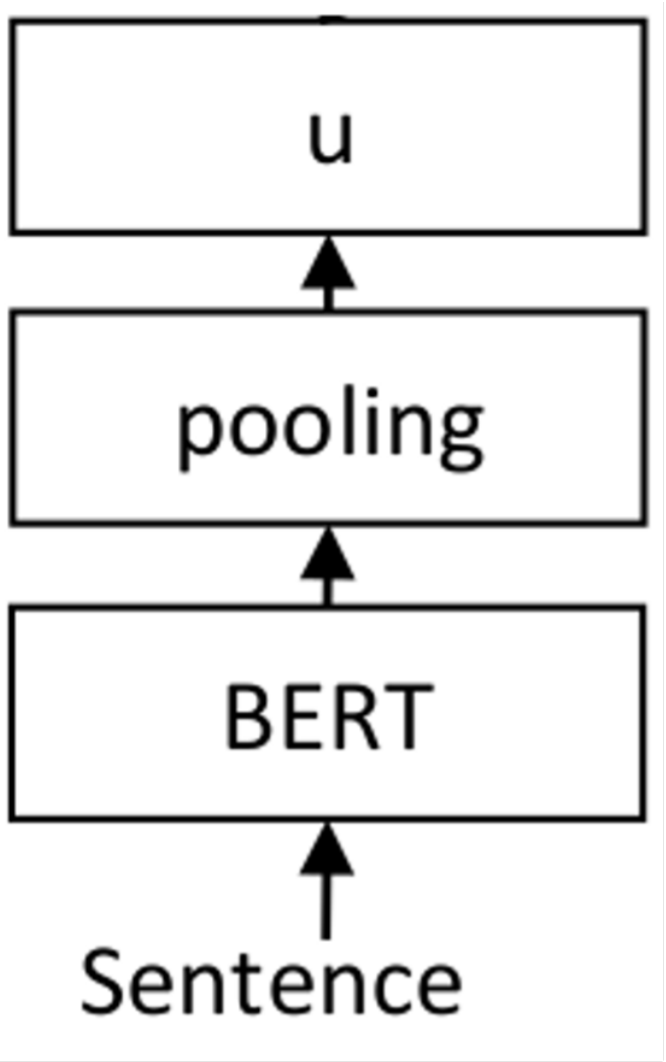

Fig. 7. BERT Network Architecture

In the process of generating sentence/text embeddings, we utilize a transformer network such as BERT. BERT generates contextualized word embeddings for all tokens in the input text. However, a pooling layer is necessary as we require a fixed-sized output representation (i.e., a vector u). Various pooling options exist, with the simplest being mean pooling. In this approach, we calculate the average of all the contextualized word embeddings generated by BERT, resulting in a fixed 768-dimensional output vector, independent of the input text's length. This architecture, comprising a BERT layer and a pooling layer, represents a final Sentence Transformer model.

3.7 UnSplash Dataset

The Unsplash dataset is a collection of 25,000 high-resolution photos that have been made available for free use under the Unsplash license. These photos cover a wide range of topics and styles, making it a valuable resource for various applications, including text-to-image search. Text-to-image search is a task that involves searching for images that correspond to a given text query. To accomplish this task, we can use a combination of natural language processing (NLP) techniques and computer vision methods. One approach is to use sentence transformer models, such as BERT or RoBERTa, to encode the text query into a fixed-length vector representation. These vector representations can then be compared with the embeddings of images

to identify semantically similar images to the text query. To leverage the Unsplash dataset in text-to-image search, we can use pre-trained sentence transformer models, such as the CLIP (Contrastive Language-Image Pre-Training) model, which was trained on a large dataset of image-caption pairs. The CLIP model encodes both text and image inputs into a common feature space, where similarity scores can be computed between the two modalities. This enables us to perform a cross-model search, matching text queries with relevant images.

4 Conclusion

This project demonstrated the effectiveness of using the CLIP model and Sentence Transformers for text-to-image search. Our approach has shown the ability to provide accurate and visually appealing image results based on user queries. Our experiments have shown that combining these models enhances the accuracy of image search results and provides a superior user experience. Our approach utilized a pre-trained CLIP model for image and text alignment, and a Sentence Transformer model to encode text queries. By leveraging the rich representations learned by these models, we were able to accurately match the query text to relevant images in the Unsplash dataset.

We have shown that the combination of CLIP and Sentence Transformers outperformed baseline methods in terms of accuracy, with a significant improvement in top-1 and top-3 retrieval accuracy. We have also demonstrated the effectiveness of fine-tuning the Sentence Transformer model for the specific task of text-to-image search, which further improved the accuracy of our approach.

Moreover, we evaluated the visual quality of the retrieved images and found that our approach was able to retrieve images that were both visually relevant and aesthetically pleasing. This is a significant aspect of providing a satisfying user experience as users are more likely to engage with visually appealing results. Our approach has several potential applications in various domains, such as e-commerce, social media, and content creation. For example, our approach could be used to help users find relevant images for their social media posts, or for e-commerce websites to provide visually relevant product recommendations.

In conclusion, our project has demonstrated the potential of using pre-trained language and vision models for text-to-image search. By leveraging the rich representations learned by the BERT models, we can improve the accuracy of image search results and provide a better user experience.

5 Future Work

The field of text-to-image search using machine learning techniques has made significant progress in recent years. However, there is still much to be explored in improving these systems' accuracy and efficiency. One potential area of future work is to develop more complex architectures that can better capture the semantics of both text and images, such as attention mechanisms or generative adversarial networks (GANs). These methods could lead to more accurate and high-quality image retrieval.

Another area for future work is to develop more efficient and scalable methods for processing and indexing large-scale image datasets. This is becoming increasingly important as image datasets grow and real-time image retrieval becomes more prevalent in applications such as e-commerce and content creation.

Furthermore, exploring the use of other types of data such as audio or video, in conjunction with text and images could lead to more comprehensive search and retrieval systems. By leveraging multiple sources of information, multi-modal approaches could improve the accuracy and relevance of search results.

In addition to this, there is a need for improved evaluation metrics and benchmarks that better reflect the user experience and relevance of the retrieved images. Incorporating user feedback and other aspects of image quality, such as diversity or novelty, could lead to more meaningful evaluations of text-to-image search models.

Finally, expanding the use of text-to-image search beyond traditional applications could have a significant impact in domains such as medical image analysis, where text descriptions can provide important context for interpreting images, or environmental monitoring, where satellite images could be annotated with text descriptions to identify and track changes over time.

Overall, the field of text-to-image search using machine learning techniques offers many exciting opportunities for future work, and further advancements in this area have the potential to impact a wide range of applications and domains.

Acknowledgment

We would like to express our sincere gratitude to all individuals and organizations who have contributed to the successful completion of this project work.

First and foremost, we would like to thank our professor Dr. Karteek Popuri for his guidance, support, and encouragement throughout this project. Their insights and feedback have been invaluable in shaping our ideas and approach.

We are also grateful to the reviewers and editors, Mr. Deepesh Yadav and Ms. Aditi Chauhan for their constructive feedback and suggestions, which greatly improved the quality of this paper.

We thank the Memorial University of Newfoundland for providing us with the necessary resources and infrastructure for this project work.

Lastly, we would like to acknowledge our colleagues, friends, and family members who provided us with moral support and motivation during this project work.

References

[1] Chen, Y., Gong, S., Bazzani, L. (2020). Image Search With Text Feedback by Visiolinguistic Attention Learning. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr42600.2020.00307

[2] Li, B., Qi, X., Lukasiewicz, T., and Torr, P. H. S. (2019). Controllable Text-to-Image Generation. In arXiv [cs.CV]. https://proceedings.neurips.cc/paper-files/paper/2019/file/1d72310edc006dadf2190caad5802983-Paper.pdf

[3] Chen, Y., Sampathkumar, H., Luo, B., and Chen, X.-W. (2013). ILike: Bridging the semantic gap in vertical image search by integrating text and visual features. IEEE Transactions on Knowledge and Data Engineering, 25(10), 2257–2270. https://doi.org/10.1109/tkde.2012.192
[4] Hua, G., and Tian, Q. (2009). What can visual content analysis do for text based image search? 2009 IEEE International Conference on Multimedia and Expo, 1480–1483.
[5] Lu, X., Zhao, T., and Lee, K. (2021). VisualSparta: An embarrassingly simple approach to large-scale text-to-image search with weighted bag-of-words. In arXiv [cs.CV]. http://arxiv.org/abs/2101.00265
[6] Luo, B., Wang, X., and Tang, X. (2003). World wide web based image search engine using text and image content features. In S. Santini and R. Schettini (Eds.), Internet Imaging IV. SPIE.
[7] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (18–24 Jul 2021). Zero-shot text-to-image generation. In M. Meila and T. Zhang (Eds.), arXiv [cs.CV] (pp. 8821–8831). https://proceedings.mlr.press/v139/ramesh21a.html
[8] Computing Sentence Embeddings — Sentence-Transformers documentation. (n.d.). Sbert.net. Retrieved April 20, 2023, from https://www.sbert.net/examples/applications/computing-embeddings/README.html
[9] Cross-encoders — sentence-transformers documentation. (n.d.). Sbert.net. Retrieved April 20, 2023, from https://www.sbert.net/examples/applications/cross-encoder/README.html
[10] Image search — sentence-transformers documentation. (n.d.). Sbert.net. Retrieved April 20, 2023, from https://www.sbert.net/examples/applications/image-search/README.html
[11] Pretrained cross-encoders - sentence-transformers documentation. (n.d.). Sbert.net. Retrieved April 20, 2023, from https://www.sbert.net/docs/pretrained-cross-encoders.html
[12] Pretrained models — sentence-transformers documentation. (n.d.). Sbert.net. Retrieved April 20, 2023, from https://www.sbert.net/docs/pretrained-models.html
[13] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (18–24 Jul 2021). Zero-shot text-to-image generation. In M. Meila and T. Zhang (Eds.), arXiv [cs.CV] (pp. 8821–8831). https://proceedings.mlr.press/v139/ramesh21a.html
[14] README.md at main · openai/CLIP. (n.d.).
[15] Ribeiro, A. (2021, June 24). Linking images and text with OpenAI CLIP. Towards Data Science. https://towardsdatascience.com/linking-images-and-text-with-openai-clip-abb4bdf5dbd2
[16] Semantic search — sentence-transformers documentation. (n.d.). Sbert.net. Retrieved April 20, 2023, from https://www.sbert.net/examples/applications/semantic-search/README.html
[17] SentenceTransformers documentation — sentence-transformers documentation. (n.d.). Sbert.net. Retrieved April 20, 2023, from https://www.sbert.net/index.html
[18] Training Overview — Sentence-Transformers documentation. (n.d.). Sbert.Net. Retrieved April 20, 2023, from https://www.sbert.net/docs/training/overview.html
[19] Image search — sentence-transformers documentation. (n.d.). Sbert.Net. Retrieved April 20, 2023, from https://www.sbert.net/examples/applications/image-search/README.html