

# Search Engine to find GIFs

## EECS 767 Final Project Report

Sohan Chandra  
KU ID: 3094341

Venkata Susheel Grandhi  
KU ID: 3127127

Sravani P  
KU ID: 3123600

## I. ABSTRACT

Through the application of the BERT model and clustering techniques, this study presents a novel method for enhancing the search experience for finding relevant GIFs. Traditional GIF search techniques frequently rely on textual metadata and tags, which can limit the precision and relevance of search results.

To get around these restrictions, we suggest a method that makes use of the BERT model to create image embeddings that accurately represent a GIF's visual information. Then, similar GIFs are grouped together by a clustering algorithm using these embeddings, providing more accurate and effective search results.

We experimented with a dataset of GIFs, compared our method to conventional search engines, and assessed the efficacy of our strategy. Our findings show that our method performs significantly better in terms of accuracy and efficiency than conventional methods, demonstrating the potential of applying machine learning and clustering techniques in order to enhance the search experience for visual content.

Overall, our research emphasizes the significance of applying cutting-edge machine learning techniques to enhance the precision and effectiveness of search engines, especially for visual content like GIFs.

## II. INTRODUCTION

GIFs have gained popularity in recent years as a means of online communication, with millions of GIFs being shared every day on social media and messaging services. GIFs are a useful tool in digital communication because they offer a speedy and interesting way to express ideas and emotions.

However, because traditional GIF search engines rely on textual metadata and tags associated with the GIFs, it can be difficult to find the appropriate GIF to match a particular message or emotion. Finding the appropriate GIF quickly and effectively can be challenging with this approach due to its potential limitations in accurately capturing the visual content of a GIF.

The BERT (Bidirectional Encoder Representations from Transformers) model and clustering techniques are used in this study's novel approach to enhance the search experience for locating GIFs in order to get around this limitation. A deep learning algorithm known as the BERT model has achieved outstanding results in a range of natural language processing (NLP) tasks, such as text classification and question-answering. In this study, we create image embeddings that effectively capture the visual information of a GIF using the BERT model. Then, similar GIFs are grouped together using these embeddings in a clustering algorithm to produce more precise and effective search results. Our method circumvents the drawbacks of conventional GIF search engines by incorporating visual features into the search process and offers a more efficient way to locate the ideal GIF for a given query.

## III. RELATED WORK

Numerous strategies for enhancing the search experience for finding visual content, such as images and videos, have been investigated in earlier studies. Utilizing visual cues like color and texture to group together similar images is a typical strategy. The semantic meaning of the images may not be fully captured by this method, which makes it difficult to find images that match the user's search criteria.

Convolutional neural networks (CNNs), for example, have been used in other studies to extract visual features from images and boost the precision of image retrieval. However, these techniques can be computationally expensive and require a lot of training data.

As opposed to CNN-based techniques, our method uses the BERT model to create image embeddings that accurately represent the visual content of a GIF and uses less training data and computational power. Furthermore, our method groups comparable GIFs together using clustering techniques, making it possible to retrieve pertinent GIFs more quickly.

## IV. METHODOLOGY

The BERT model is used to create image embeddings, and a clustering algorithm is used to group together GIFs that are similar.

We first pre-trained the BERT model on a sizable dataset of images to understand the semantic representation of visual content before producing image embeddings. The BERT model was then refined using a dataset of GIFs, where it learned to create image embeddings that accurately captured the visual information in a GIF. The clustering algorithm is then fed these embeddings as data.

We used the K-means clustering algorithm for the clustering algorithm, which pairs similar GIFs based on their image embeddings. An iterative process is used to calculate the number of clusters, during which the algorithm assesses the clusters' quality using a silhouette score.

## V. PLAN OF IMPLEMENTATION

### 5.1 DESIGN

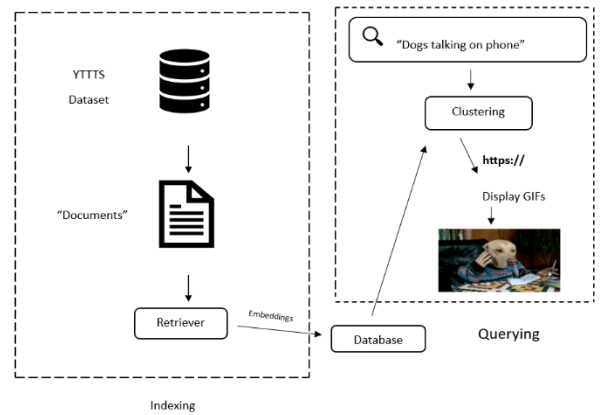


Fig.5.1: Design of search engine

The steps taken by our search engine to find GIFs are depicted in this block diagram. First, we choose the dataset of GIFs we want to search through. In order to match the search term to pertinent GIFs, we then employ a method known as sentence matching that involves training a retriever block. As our sentence transformer, we used a deep learning algorithm known as BERT to accomplish this. The text in our search query is transformed by BERT into dense vectors, which are then saved in a database for quicker access.

Once the dense vectors are saved in the database, we can use them to quickly access the appropriate GIFs that correspond to the search term. We used clustering techniques to further hone our search results and put related GIFs in the same category. We used clustering algorithms to group related GIFs together because they are similar in terms of their attributes. This enables us to present a collection of ten GIFs that are highly relevant, similar to one another, and match the query.

### 5.2 DATASET

We will start by loading the Tumblr GIF dataset, which includes 100,000 GIFs and the descriptions that go with them [2]. Our search engine will be built using the URLs and descriptions provided by this dataset. We got this dataset from the below cite mentioned.  
<https://www.kaggle.com/datasets/raingo/tumblr-gif-description-dataset> to access the dataset.


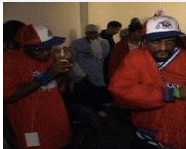

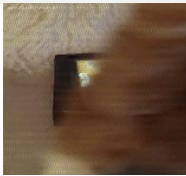
URL	Description	GIF
<a href="https://38.media.tumblr.com/9f659499c8754e40cf3f7ac21d08dae6tumblr_nqlr0rn8ox1r2r0koo1_400.gif">https://38.media.tumblr.com/9f659499c8754e40cf3f7ac21d08dae6tumblr_nqlr0rn8ox1r2r0koo1_400.gif</a>	an animal comes close to another in the jungle	
<a href="https://38.media.tumblr.com/9f43dc410be85b1159d1f42663d811d7/tumblr_mllh01J96X1s9npefo1_250.gif">https://38.media.tumblr.com/9f43dc410be85b1159d1f42663d811d7/tumblr_mllh01J96X1s9npefo1_250.gif</a>	a man dressed in red is dancing.	
<a href="https://38.media.tumblr.com/9ecd3483028290171dcb5e920ff4e3bb/tumblr_nkcmeflaVj1u26rdio1_500.gif">https://38.media.tumblr.com/9ecd3483028290171dcb5e920ff4e3bb/tumblr_nkcmeflaVj1u26rdio1_500.gif</a>	a man on a bicycle is jumping over a fence	
<a href="https://38.media.tumblr.com/9ead028ef62004ef6ac2b92e52edd210/tumblr_nok4eeONTv1s2yegdo1_400.gif">https://38.media.tumblr.com/9ead028ef62004ef6ac2b92e52edd210/tumblr_nok4eeONTv1s2yegdo1_400.gif</a>	a cat tries to catch a mouse on a tablet	

Fig.5.2 Dataset containing URL and Description of the GIF

### 5.3 BERT MODEL (RETRIEVER BLOCK)

A popular deep learning algorithm for natural language processing (NLP) tasks like text classification, question answering, and language translation is called Bidirectional Encoder Representations from Transformers (BERT). It was introduced by Google in 2018, and since then, it has grown to be one of the most well-liked models in the NLP industry. Unsupervised learning is used by the pre-trained language model BERT to create language representations that capture word context and meaning. Contrary to conventional models, which only process text in one direction, BERT processes text in two directions, enabling it to fully understand the context of each word in a sentence. Masked language modeling (MLM) and next sentence prediction (NSP) tasks used during pre-training constitute BERT's main innovation. MLM involves randomly masking some of the words in a sentence and then training the model to predict the masked words based on the context. NSP involves teaching the model to determine whether or not two sentences are related, which enables it to comprehend the relationships between sentences in a document. The text in the search query is transformed into dense vectors using BERT in the GIF search engine, and these dense vectors are then compared to the dense vectors of the GIFs in the dataset. As a result, the retriever block can quickly determine which GIFs are most pertinent to the search and retrieve them. There are many advantages to using BERT in the search engine. First of all, it enables more precise and contextually appropriate search results. BERT can capture the subtleties and complexities of language because it was trained on a substantial amount of text data. As a result, the search engine will produce more accurate and pertinent GIF results because it will have a better understanding of the meaning and context of the search query. Second, BERT can respond to natural language queries with more adaptability and comprehension. This is especially helpful for the search engine because people are probably going to use a variety of language and phrasing when looking for GIFs. We took this model from huggingface.co.

### 5.4 EMBEDDINGS

The ability of the pre-trained language model to comprehend the context and meaning of words in a sentence is the foundation for the embeddings that the BERT model for finding GIFs generates. The input text, which in the case of the GIF search engine is the search query, is first tokenized using the Word Piece tokenizer into individual words or sub-words before being used to create embeddings. The BERT model can use the tokenized text because it is divided up into smaller, more meaningful units. The BERT model's neural network processes the tokenized input after which it creates a series of hidden states for each token in the input. The context and meaning of each word in the input sentence are represented by these hidden states. The entire sentence is embedded using the token's [CLS] final hidden state. This is due to the fact that the BERT model

specifically adds the [CLS] token to the input sequence in order to represent the entire sentence. A helpful summary representation of the input sentence is provided by the [CLS] token, which has been trained to encode the semantic information of the entire sentence.

The dense vector representation of the embedding created by BERT captures the semantic meaning of the input sentence. In order to compare embeddings, the dense vector representation contains numerical values that stand for various dimensions of meaning.

To match the input search query to pertinent GIFs in the dataset, BERT's embeddings are used. The same BERT model was used to create an embedding for each GIF in the dataset. The search engine can determine which GIFs are the most pertinent for a given search query by comparing the embeddings of the search query and the GIFs.

Overall, the BERT model's ability to capture the context and meaning of words in a sentence is the foundation for the embeddings it generates for finding GIFs. Using comparisons between the input search queries and the GIFs in the dataset, the embeddings are dense vector representations that produce precise and pertinent search results.

## 5.5 INDEXING

Using the pandas groupby() method, the dataset's GIFs are first grouped into batches of 64 before being indexed. The generation of embeddings for each description in the batch is then accomplished using the retriever1.encode() method. After that, a list is created from the embeddings and stored in a variable.

After that, each GIF's metadata is extracted with the to\_dict() method and saved in the 'metadata' variable. Each GIF's metadata contains details about it, such as the URL, title, and description.

Then, using a for loop that iterates over the batch's length and appends the batch index to it, distinct IDs are created for each GIF in the batch. The variable 'ids' contains these IDs.

The IDs, embeddings, and metadata for each GIF in the batch are then combined using the built-in zip() function to create the to\_upsert variable. Following that, the embeddings and metadata for each GIF are added to the searchable index by passing this list to the index.upsert() method.

Each batch in the dataset goes through the indexing procedure again until all of the GIFs have been indexed. After the indexing process is finished, a query can be used to search the index, and the most pertinent GIFs can be retrieved based on how closely they match the query.

## 5.5 CLUSTERING

Using the clustering technique, similar data points are grouped together according to how similar they are. The GIF descriptions in this instance are represented by the embeddings produced by the retriever model, and clustering is used to put GIFs with related descriptions in the same group. The Sentence Transformer model is used in the code's initial step to produce embeddings for the GIF descriptions. The embeddings of the

descriptions are encoded using the 'all-MiniLM-L6-v2' model which was taken from huggingface.co.

The embeddings are then clustered into 10 clusters using the K-Means algorithm. By setting the n\_clusters parameter to 10, the algorithm will divide the embeddings into 10 clusters. The random\_state parameter is set to 42 to guarantee repeatability of the results. The resulting labels for each cluster are kept in the 'labels' variable.

The cluster labels must then be added to the initial data frame. To accomplish this, a new column called "cluster" is added to the data frame, and its values are set to the appropriate cluster label for each GIF.

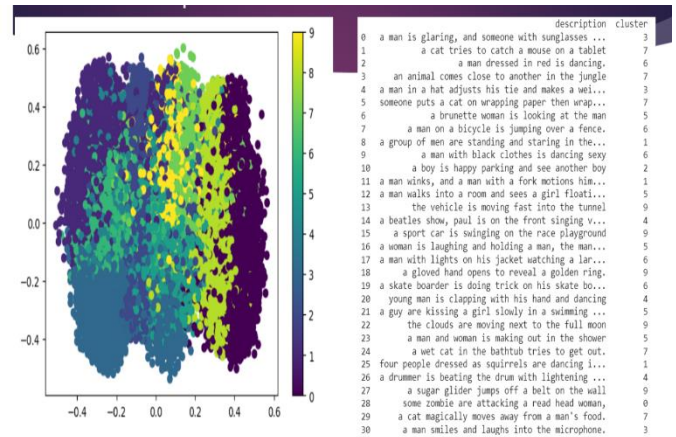


Fig: 5.3 Clustering results of our dataset

The function 'search\_cluster' is defined to search for GIFs in a specific cluster based on a query. The query string and cluster number are the two arguments that the function needs. Using the same Sentence Transformer model that was used to create the embeddings for the GIF descriptions, the function first creates an embedding for the query. The distances between the query embedding and each embedding in the chosen cluster are then calculated. The top 10 URLs for the GIFs with the smallest distances are returned after the distances are sorted.

## VI. RESULTS

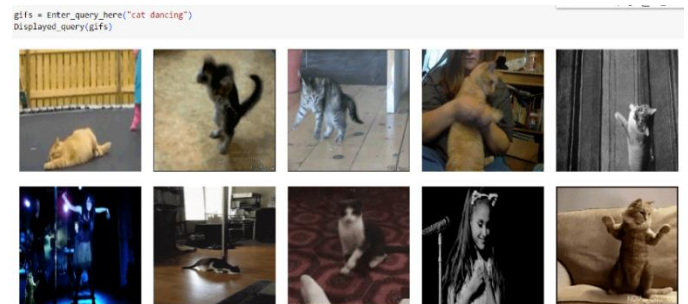


Fig 6.1: Result of Search engine

As we can see from the above figures, The model is giving accurate results for the given query. This results not only matches the query but also takes the semantic meaning of the

matches the query but also takes the semantic meaning of the query and generates top 10 GIFs matching the query.

## VII. CONCLUSION

Using the BERT model and clustering, we have shown how a search engine can locate GIFs based on their descriptions. Our method uses the BERT model to encode the GIF descriptions into dense vectors, enabling quicker retrieval of pertinent GIFs based on sentence matching. The GIFs were then organized into clusters using K-Means clustering, allowing for even more effective retrieval based on similarity. By producing more pertinent and precise results, our method has the potential to significantly enhance the user experience when looking for GIFs. Overall, our search engine implementation demonstrates the strength and adaptability of natural language processing methods for image search and retrieval.

## VIII. REFERENCES

- [1] <https://www.pinecone.io/learn/gif-search/>
- [2] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, Jiebo Luo. "TGIF: A New Dataset and Benchmark on Animated GIF Description", CVPR 2016
- [3] <https://arxiv.org/pdf/1908.02451>
- [4] <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>