SI 650 / EECS 549 Final Project Report
Artwork Search Engine

Zhiyuan Zhang, Cheng Qian

December 14, 2021

1 Introduction

The Metropolitan Museum of Art is located at the Fifth Avenue and Fort Tryon Park in New York City. It collects artworks that span over 5000 years from all over the world [1]. Now, high-resolution images of over 406 thousand pieces of artworks are being shared on The Met website, which is open to use to artists, designers, educators, students and so on [2].

Exploring in the world of art is fun. However, there are too many artworks, from different classification, culture, periods... And the existing search function of the website is not so great. It always returns some unrelated items, which makes it hard for people to find the artworks they are interested in. For example, when I search 'Chinese paintings', many completely unrelated objects appear in the first page, such as porcelain and furniture. In addition, for not famous artworks, their textual description is very short. Incomplete information increases the difficulty of searching for these works. Also, there are very few researches of vertical searching on artworks. Thus, we want to build a vertical search engine on the paintings and drawings in THE MET.

Some previous works use traditional search which looks for exact lexical matches of the query words in documents. This method cannot handle synonyms or some variations of different parts of speech while the wording used in the introduction of the artwork has some characteristics. We have proposed 2 new methods use to handle these problems: learning to rank with designed features and dense embedding with language models. Our approaches utilize the different features or the contextual information of documents. The results of our model exceed BM25 baseline we built before. It shows better ability in artwork search.

Our designed search engine can be used on the museum's website. Visitors can find works of interest to them and get relevant information, such as the location of this work in the museum, which can help them visit the museum more conveniently. It can also be used on the art trading platform to help users to select artworks easily.

Our contribution includes: (1) We utilize the trendy methods to deal with synonyms, acronyms or word variations in queries and documents. (2) In learning to rank method, we designed some distinguishable to better determine the relevance. (3) In dense retrieval method, we extracted the contextual information from words and represented the documents and queries in the same dense vector space.

2 Data

The Metropolitan Museum of Art provides selected datasets of information on more than 470,000 artworks in its Collection for unrestricted commercial and noncommercial use. The data we will use can be found on Github [1]. It's uploaded by The Met and open to public.

The dataset is too large to deal with, so that we only focus on paintings and drawings in the collection. Also, There are many duplicate artworks which are exactly the same. We filter out these duplicated items by comparing whether the information of 2 item is identical. After filtering, there are 34426 artworks left.

For each artwork, it has 43 features such as object ID, name, title, period, artist info, etc. It also provides a link directing to the detail page on the museum's website. On the detail page of each

[1] https://github.com/metmuseum/openaccess

artwork, it shows a paragraph of artwork description. We collect these descriptions and add them to the feature of the objects by web scraping. Table 1 shows 2 examples of the instances with some selected features.

Object ID	Title	Artist Name	Period	Description
35986	A Rooster near Trees	Cheng Jiasui	Ming (1368–1644)–Qing (1644–1911) dynasty	/
438004	Poppy Fields near Argenteuil	Claude Monet	/	This work is one of four similar views.....
436524	Sunflowers	Vincent van Gogh	/	Van Gogh painted four still lifes of sunflowers.....

Table 1: Examples of instance with several features

In dense retrieval, we simply concatenate them into a single string to form a (object id, text) pair for each instance. Since usually the title and artist is more important than other information. We repeat the title and artist name 3 times to emphasize.

There are no ground truth relevance in the dataset, so we manually marked the relevance of about 100 artworks each query for 20 queries in 5 point scale. (1 means not related, 5 means exactly match) We didn't do it randomly. We rated the documents that are regarded as 'related' by some search engines. We first built several simple search engines with different score functions, then we retrieved 50 documents for each query on these search engines and the original searching system on the MET's website. We have tried SimpleSearcher of Pyserini and score functions in HW2. Then we rated these retrieved documents. In addition, we did some query-related search, looking for some other documents and rating them.

Here 2 are some statistics information about the rating with the queries we did, such as the relevance score distribution.

Relevance \Query	sunflowers paintings	Ming dynasty moutain landmark paintings	Chinese horse paintings
Labeled instances	102	121	101
1 (least relevant)	18.6%	6.78%	8.9%
2	22.6%	10%	17.8%
3	42.07%	40.5%	30.7%
4	8.82%	17.35%	11.9%
5 (most relevant)	6.86%	25.44%	30.65%

Table 2: Relevance distrubution of labeled instances for 3 queries

3 Related Works

Several approaches have been used to retrieve artworks. Yelizaveta et al. used image processing skills to retrieve artworks [13]. They focused on paintings exhibits particular color patterns, such as "paintings in warm colors" [13]. Kadocura and Osana's approach retrieve information based on similarity of touches[6]. Zirnhelt and Breckon's approach combined color and texture similarity to retrieve artwork[14]. Kangying et al. proposed a framework utilizing multi-lingual, multi-modal information embedding framework for Ukiyo-e record retrieval utilizing pre-trained language nad image model like BERT and ImageNet[9]. Mao et al. proposed a model that learns joint representation of visual arts, which can be potentially integrated into information retrieval[10]. While previous attempts mainly

focused on image processing skills, our approach focus on more text analysis utilizing information such as period, region, description and so on.

For textual information, there are 2 main ways to implement the retrieval models. [12] has summarized many tf-idf based retrieval functions such as BM25, Pivoted, Divergence from Randomness Models and their variants, etc. Also, [7] shows that different BM25 have very close effectiveness. There are not many previous works based on the retrieval of artworks. The texts of artworks are a little special since many of them are very short within several words. We would like to improve the existing BM25 models and fine tune them to fit the artworks. Another way is to do IR with dense vectors. To handle synonymy, instead of (sparse) word-count vectors, many people use (dense) embeddings. This idea was proposed quite early with the LSI approach [5]. Some methods use other ways to represent the encoded text, such as using average pooling over the BERT outputs of all tokens instead of using the CLS token, or adding extra weight matrices[8].

4 Methods

4.1 Learning to Rank

Learning to rank is a machine learning application in information retrieval. We use a pipeline for our learning to rank model. First, we utilize BM25 to retrieve artworks based on their descriptions. Then, we chain the BM25 with a set of features. The features we include are:

- score of BM25 after sequential dependence and query expansion
- score of Dirichlet Language Model on the title of artworks
- score of Coordinate Match on the artist name
- score of Dirichlet Language Model on the tags of artworks
- score of Dirichlet Language Model on the origin country of artworks
- score of Dirichlet Language Model on the department that artworks belong to
- score of Dirichlet Language Model on the culture that artworks belong to
- score of Dirichlet Language Model on the Period the artworks are in
- whether the artwork is popular and important in a collection
- whether the artwork is on the Timeline of Art History Website
- score of Dirichlet Language Model on the description of artworks

It is common to include ranking scores like BM25. The first feature in our set not only uses BM25, but also rewrite queries using sequential dependence model and performed query expansion to increase the likelihood of matching relevant artworks. In addition, we also consider the origin of artworks, their periods, titles, tags, the department and culture they are from and whether they are important as an artwork, which are some properties that people care most. Then, we chain these features to a LambdaMART model as our learning model. LamdaMART is a supervised machine learning algorithm that gives ordering of items. It is a pairwise approach. It uses LambdaRANK as its cost function and uses gradient boosted decision tree to predict results. It is proved to have better performance than the original RankNet [4].

4.2 Semantic Search With Language Models

Information of an artwork can be divided into textual data in natural language and its image. For the textual data, some artworks only have titles and artists' names without long description. For these artworks, the textual data is very short and incomplete. The penalty to long document's score isn't very reasonable. Because of the complexity of the contents in documents, we decide to use dense embeddings to handle these problems.

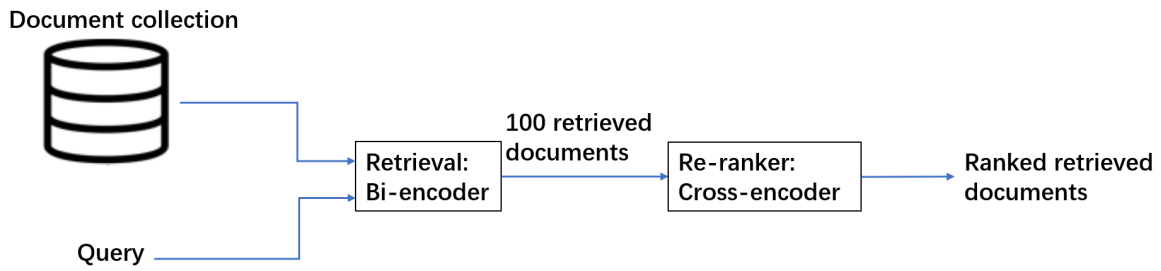

Figure 1: Pipeline of our retrieval system.

The pipeline of our retrieval system has the following components: bi-encoder and cross-encoder.

For the bi-encoder, we decide to use semantic search (dense retrieval) to encode the queries and documents. Unlike traditional retrieval technique, semantic search doesn't depend on lexical matches. Instead, it understands the contextual contents and it can represent 2 similar contexts into 2 vector embeddings which is close on the vector space without the problem of synonyms.

Here, we use a pre-trained model *msmarco-distilbert-base-v2* from *Sentence-Transformers* as the bi-encoder [11]. This is a general language model trained on user queries using Bing. We first use this model to encode and store all the documents in the collection in advance. Given a query, we use the same model to represent it in the same vector space. Then we can compare the similarity between each document and the query. This is implemented by the function *semantic_search* from *Sentence-Transformers* [11]. We select the first 100 retrieved documents as candidates which may be relevant to the query.

However, the retrieval system with a single encoder may not be enough. Thus, we use a re-ranker based on cross encoder to re-rank the documents retrieved in the previous stage. Here we use pre-trained *cross-encoder/ms-marco-MiniLM-L-6-v2* from *Sentence-Transformers* as the cross-encoder [11]. We concatenate each document and the query respectively as the input to the cross-encoder. It can perform transformer attention across the document and the query to determine the attention between terms. The output is a single value which indicates the degree of relevance between a document and a query. We use these scores to re-rank the documents.

In our implementation, we implement the retrieval using BM25 [3], a single bi-encoder and the combination of bi-encoder and cross-encoder. We evaluate and compare their performance using the same queries. These are discussed in the evaluation section.

5 Evaluation and Results

5.1 Baseline

For evaluation, we plan to compare our result against random performance and some basic BM25. The queries come from our selected queries with annotated relevance described in data section. We select 3 queries. We are using NDCG@k where k=5 as the metrics.

The first baseline is the random performance. For the searching result, we randomly select 5 documents from the collection. Since the collection is very large and the number of related items is very small, there is no doubt that the NDCG is 0.

The second baseline is the BM25 model we built in homework2 with parameters k1=0.45, b=0.55, k3=6. The sample resulting NDCG of the BM25 search engine is shown in table 3. The NDCG of the first query is 1 because there are only several artworks are related with 'sunflowers'. I annotated all these several matched artworks 5.

From the retrieval results, the $NDCG_5$ for query "sunflowers paintings" is 1. The $NDCG_5$ for

Query	NDCG at k=5
sunflowers paintings	1
Ming dynasty mountain landscape paintings	0.305
Chinese horse paintings	0.285

Table 3: NDCG at k=5 of 3 queries for BM25

query "Ming dynasty mountain landscape paintings" is

$$\begin{aligned} NDCG_5 &= \frac{1 + 5/\log_2 2 + 3/\log_2 3 + 4/\log_2 4 + 5/\log_2 5}{5 + 5/\log_2 2 + 5/\log_2 3 + 5/\log_2 4 + 5/\log_2 5} \\ &= 0.305 \end{aligned}$$

.The $NDCG_5$ for query "Chinese horse paintings" is

$$\begin{aligned} NDCG_5 &= \frac{5 + 3/\log_2 2 + 1/\log_2 3 + 1/\log_2 4 + 5/\log_2 5}{5 + 5/\log_2 2 + 5/\log_2 3 + 5/\log_2 4 + 5/\log_2 5} \\ &= 0.285 \end{aligned}$$

5.2 Evaluation of Learning to Rank

Here, we compare our learning to rank model with the baseline BM25. The metric we choose is NDCG. The result is shown in Table 4. From Table 4, we see that the Learning 2 Rank model outperforms

name	ndcg_cut_5	ndcg_cut_10	ndcg_cut_20	ndcg
BM25	0.3229	0.3103	0.2843	0.2564
BM25 + Learning2Rank	0.4871	0.3913	0.3286	0.2828

Table 4: Comparison of NDCG between Learning2Rank and BM25

BM25 on NDCG@5, NDCG@10, NDCG@20 and NDCG. The improvement is huge on top ranks, which confirms our thoughts that features increase the accuracy of top results. Because the model pays attention to different features by giving them different weights. The retrieval process returns finer results. But it also takes longer to retrieve, but it's still less than 2 seconds. Compared with top results, the improvement on NDCG over all artworks retrieved is smaller. The reason is probably that the relevant artworks in the dataset are almost all retrieved, so the result is close.

5.3 Evaluation of Dense Retrieval

Here we compare NDCG@5 of our proposed model against others. We choose 5 queries which come from our annotation to evaluate our system. BM25 model here is implemented by the function in *rank_bm25* [3] library. Table 5 shows the result of retrieval with 5 queries. We can see that generally NDCG@5 of bi-encoder and cross-encoder retrieval are much better than BM25 in most cases. Cross-encoder is a little better than bi-encoder i the results. This shows that our implementation has a better performance than an untuned BM25.

There are some hyperparameters we can adjust in the model. For example, we can decide the number of candidates retrieved in the bi-encoder retrieval. I have tested the result with top_k=50,100,150,200, and find 100 is a better choice. When top_k=50, some actually related documents may be missed. When it equals to 200, it has few effects compared to 100. Some irrelevant documents will also be selected which may act as noise.

The encoding models we choose also affects. Generally, a more complex model trained on a larger corpus has a better performance.

Queries	NDCG@5 of BM25	NDCG@5 of Bi-encoder retrieval	NDCG@5 of Cross-encoder retrieval
Chinese landscape with mountain and river	0.252	0.895	0.852
harlem renaissance	0.677	0641	0.760
Chinese calligraphy of Song dynasty	0.492	0.594	0.615
abstract expressionism	0.827	0.761	0.847
Neo Impressionism and Pointillism	0.781	0.677	0.764

Table 5: Comparison of NDCG@5 among different methods

6 Discussion

6.1 Learning to Rank

In the evaluation section, we see that the learning to rank model outperforms BM25. The reason is that the LambdaMART model weighs different features and consider them when ranking. So, the returned result is more accurate in top rankings. The improvement is consistent as far as the top 30 results. However, the result is close to BM25 over top 50 results. One reason may be that most relevant artworks are retrieved in both approaches. Another reason is that the increase of retrieved document number normalizes the result.

One improvement that can further improve the performance and make the model more robust is to increase the number of annotations so the model can learn more and is less likely to overfit.

6.2 Dense Retrieval

As shown in Table 5, the results of our model outperforms the baseline BM25 in most cases which are expected. From my own experience, our search engine performs better than the existing search engine on the website of THE MET. This search engine can give users better experiences when searching artworks. It returns better results.

The main difference between our method and BM25 is that we use contextual dense representation instead of looking for lexical matches. Take Query 1 *"Chinese landscape with mountain and river"* as an example, many high-rank documents retrieved by BM25 belong to other cultures such as Japan or Europe. It only focus on the exact match of words while it doesn't consider the semantic integrity in the query. This explains why NDCG for this query is very low. Also, the description of artwork may use some different words with similar meaning to the query word. This kind of documents are hard to retrieve as well. In addition, BM25 is not suitable for SE for artworks because we trend to give penalty to artworks with very short description and rank the documents with long descriptions higher. Users prefer to see famous works with more information.

Instead, our dense retrieval system can handle synonyms and the semantic integrity at the same time. It can take care of all the elements in the sentence and the document length is not an influencing factor. This is the main reason that dense retrieval outperforms BM25.

However, in some cases the dense retrieval doesn't perform better than BM25 such as the query *"Neo Impressionism and Pointillism"*. This may due to the incompleteness of the annotations. We only annotate 100 documents for each query and let the relevance score of other documents be 1 (not relevant). Our search engine may retrieve some actually relevant documents not annotated which lowers NDCG for all of BM25, bi-encoder and cross-encoder retrieval. Another point is that the encoders we use are pre-trained for general NLP tasks. We have not fine tune the models to fit artwork datasets well. Also, we only considered English and not multilingual while there are other languages like Chinese and Spanish in the documents. It is hard for our language models to understand multilingual information.

7 Conclusion

We have implemented artwork search engine based on learning to rank and dense semantic retrieval. Both methods show a better performance than an untuned BM25. Because of the special nature of the words of the artwork, machine learning based methods are more suitable generally. There are some potential directions for future study: annotate more samples with more queries and fine-tune the language model for dense retrieval. Our codes for this work can be found on Github: https://github.com/zhiyuan-Z/SI650-Project

8 Other Things We Tried

We have tried to do image search which return similar images given the query image in the beginning. Images of each artwork are scrapped from MET's websites as well. We use deep learning techniques to use a pre-trained neural network as local feature extractor. It produces a global and compact fixed-length representation for each image. All these representations are calculated and stored in advance. When a user input a query image, we make a representation of this image use the same model. Then we compare this representation with all the image representations to do ranking.

But we find this method is not very feasible for us because of the massive amount of computation. There are tens of thousands of high-resolution images we need to convert. Also, some artworks have tens of images and some have no image at all. It is difficult to decide how to balance all the images and how to deal with no-image artworks. Thus we give up this idea.

9 What We Would Have Done Differently or Next

For future work, we would like to try multilingual models to enable search in other languages. We also want to fine-tune the language models using the artwork datasets although this may be computationally expensive. In addition, since users prefer to see famous works with more information, We would weigh the documents' score based on their amount of information such as the length of description. We would also try to combine the machine learning and semantic search methods but still maintain a reasonable computation load. We would also integrate image representations into our search engine.

References

[1] About the met.

[2] Open access at the met.

[3] Dorian Brown. Rank-bm25: A collection of bm25 algorithms in python, 2020.

[4] Chris J.C. Burges. From ranknet to lambdarank to lambdamart: An overview. Technical Report MSR-TR-2010-82, June 2010.

[5] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.

[6] Kento KADOKURA and Yuko OSANA. Image (artwork) retrieval based on similarity of touch by self-organizing map with refractoriness. *Proceedings of NOLTA, Luzern*, 2014.

[7] Chris Kamphuis, Arjen P. de Vries, Leonid Boytsov, and Jimmy Lin. Which bm25 do you mean? a large-scale reproducibility study of scoring variants. In Joemon M. Jose, Emine Yilmaz, João Magalhães, Pablo Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins, editors, *Advances in Information Retrieval*, pages 28–34, Cham, 2020. Springer International Publishing.

[8] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering, 2019.

[9] Kangying Li, Biligsaikhan Batjargal, Akira Maeda, and Ryo Akama. Artwork information embedding framework for multi-source ukiyo-e record retrieval. In Emi Ishita, Natalie Lee San Pang, and Lihong Zhou, editors, *Digital Libraries at Times of Massive Societal Transition*, pages 255–261, Cham, 2020. Springer International Publishing.

[10] Hui Mao, Ming Cheung, and James She. Deepart: Learning joint representations of visual arts. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, page 1183–1191, New York, NY, USA, 2017. Association for Computing Machinery.

[11] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[12] Peilin Yang and Hui Fang. A reproducibility study of information retrieval models. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, ICTIR '16, page 77–86, New York, NY, USA, 2016. Association for Computing Machinery.

[13] M. Yelizaveta, Chua Tat-Seng, and A. Irina. Analysis and retrieval of paintings using artistic color concepts. In *2005 IEEE International Conference on Multimedia and Expo*, pages 1246–1249, 2005.

[14] Stephanie Zirnhelt and Toby P. Breckon. Artwork image retrieval using weighted colour and texture similarity. In *4th European Conference on Visual Media Production*, pages 1–1, 2007.