Learning Similarities: An Ensemble Model for Textual Query Image Retrieval System

James Yeh, Eric Chien, Siyadong Xiong*
Team Ensemble of Weak Students
Cornell Tech
Cornell University
New York, NY, 10011
{cy443, jc3256, sx225}@cornell.edu

Abstract

In this paper, we addressed the task of cross-modal retrieval problem specifying in using natural language query to retrieve related images. We utilized image features extracted by state-of-the-art deep learning techniques and combined them with natural language processing methods. We employed SVMs to map query vectors to object tag space, PLS regressions to find a common sub-space for query vectors and image features,, and proposed an ensemble meta algorithm to blend our models. Experimental results demonstrated that our proposed approach effectively utilized cross-modal information and explored their underlying relation, outperforming other teams on Kaggle competition.

1 Introduction

Over decades, the image retrieval problem received lots of attention in both academia and industry [1]. Researchers utilized variant approaches making retrieval systems reach a high level of maturity. Specifically image retrieval using natural language query aims to discover underlying relations between textual query and images and compute their similarity using extracted features.

Recently, Deep Learning, with the development of computational capability and scale of data set, has gained remarkable progress in many research areas such as Computer Vision and Natural Language Processing. For example, the ResNet proposed by K. He [2] reached significant results in the ImageNet competition in 2015.

1.1 Problem definition

As part of the final exam for CS 5785 Applied Machine Learning at Cornell Tech, we built a image search engine that returns a list of images based on natural language text queries.

Data Each group of students are given 10,000 samples of 224x224 JPG images for training. All images come with a list of tags describing the objects appearing in the image, a five-sentence description, and feature vectors with features extracted from pool5 and fc1000 layer of ResNet.

Evaluation For each description, students are to submit 20 candidates, and the system will evaluate the results using the MAP@20 metric, returning the score based on the ranking of the correct image.

*James, Eric, and Siyadong are respectively M.Eng. in ORIE, M.S. in IS and M.Eng. in CS student at Cornell Tech.

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

1.2 Related Work

The fundamental task of an image search engine is to take one type of data as the query, and retrieve relevant data of another type. In other words, it needs to learn and grasp the relationship among various modalities presented by text and images separately. A number of papers have provided us insights on the pros and cons of different approaches to solving this challenge.

Firstly, *A Comprehensive Survey on Cross-modal Retrieval* [5] gave us an overview of various methods used in cross-modal retrieval, including shallow methods, deep methods and binary representation methods.

In *Combined Regression And Ranking* [7], the authors proposed an efficient and effective Combined Regression and Ranking method (CRR) that optimizes regression and ranking objectives simultaneously.

While in *A scalable re-ranking method for content-based image retrieval* [1], the author argues that most Content-based Image Retrieval (CBIR) systems consider only a pairwise analysis, measuring only the similarity between pairs of images, and ignored the rich information encoded in the relations among several images. Therefore, re-ranking methods have been proposed to exploit the contextual information and improve the effectiveness of CBIR systems. The method relies on the similarity of top-k lists produced by efficient indexing structures, instead of using distance information from the entire collection.

The authors of *Continuum Regression for Cross-modal Multimedia Retrieval* also pointed out that canonical correlation analysis (CCA) and its variants may cause information dissipation when switching the modals, and explained the benefit of the use of continuum regression (CR) model to handle this task.

2 Model architecture

Fig.1 shows our final model. In reaching this architecture, we have tried different approaches to analyzing the underlying relations between text and images.

The problem was broken into three main components: 1). mapping input descriptions to tags, 2). mapping image features to input descriptions and 3). combining the result of the aforementioned components to produce the final output.

2.1 Mapping input descriptions to tags

Upon inspecting the given data, we quickly realized mapping description to existing tags is the obvious problem to target first. Both descriptions and tags are in the same domain and the correlation between the two are pretty clear. To formulate this succinctly, we want to find a function:

$$F(BoW_{description}) = BoW_{tags}$$

2.1.1 First attempt - Cosine similarity of two word vectors

In this attempt, the input descriptions are converted to to a Bag of Words (BoW) representation using the 80 categories of the tags as dictionary. Using these word vectors we compare all pairwise similarities using cosine similarity as the metric.

$$cos(x, y) = \frac{x \cdot y}{||x|| \cdot ||y||}$$

2.1.2 Second attempt - Handcrafted word conversion code book

To improve upon the previous result, we examined which tags are doing poorly and realized that for example, images with the tag "person", often do not have person as a word inside their corresponding descriptions. To fix this, we inspected the top words for each tag and then created a code book to convert these top words into their corresponding tags in the preprocessing stage. E.g. Man -> Person.

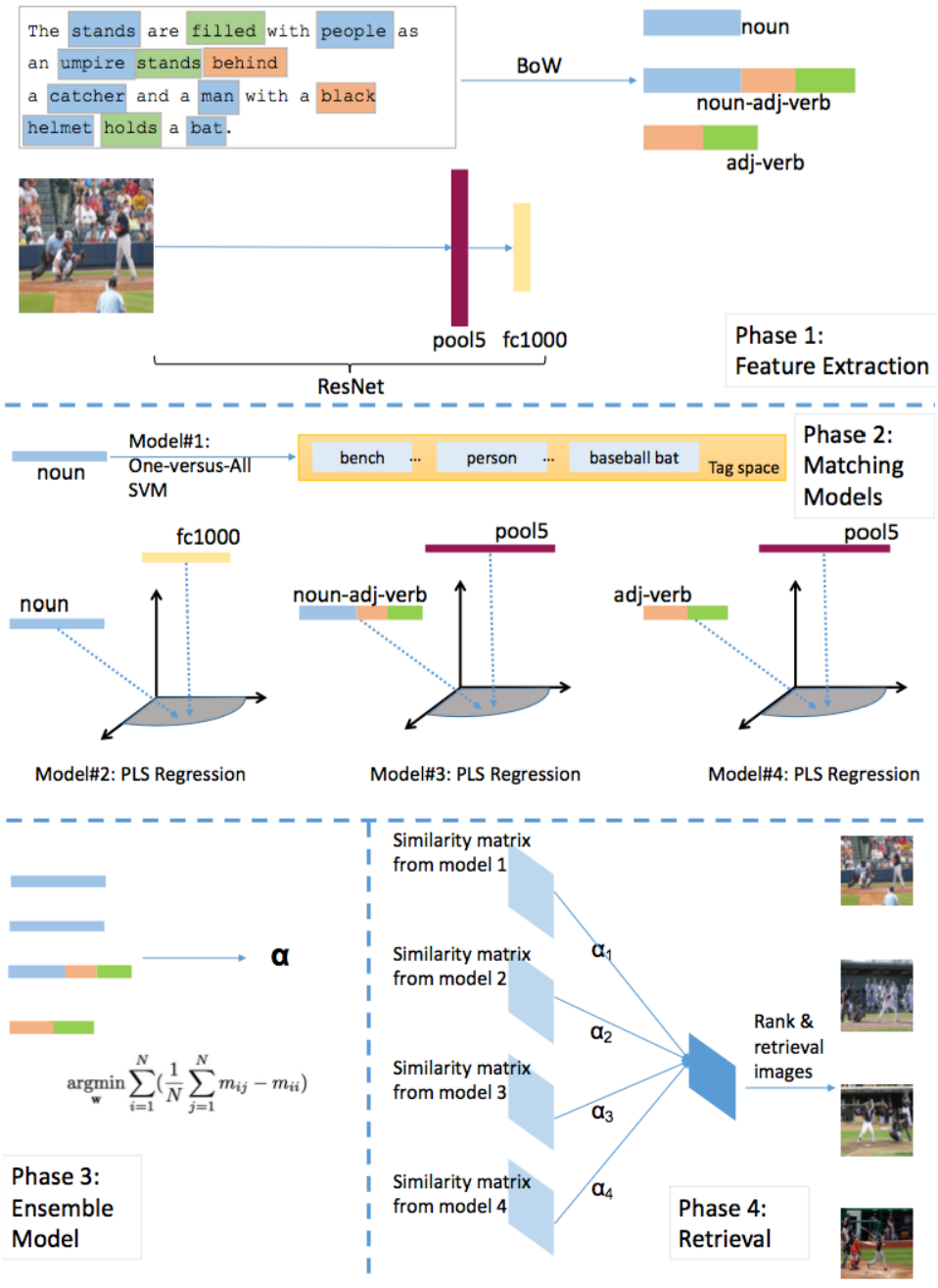

Figure 1: Model structures.

2.1.3 Final version - Training the correspondence from description to tags

After the aforementioned change is made, we began looking for ways to further improve the performance. After doing some investigation, we found that the tag file often contains more tags than it is found using description; that is,

$$|BoW_{tags\_from\_descriptions}| \leq |BoW_{tags}|$$

The main reason for this is that the descriptions often than not fails to describe the whole image in detail. For example, an image with description "A car driving pass an intersection", does not explicitly state there is traffic light in the image. To be able to infer these latent tags hidden in the description, a model mapping $BoW_{description}$ to BoW_{tags} is trained. We trained a Support Vector Machine with Linear Kernel using "One vs. All" strategy. Another benefit of this model that no handcrafted code book is needed this eliminate the need of human intervention in developing the system.

One preprocessing worth noting is that, since we are mapping to 80 nouns only, when creating $BoW_{description}$ we only consider nouns and we set the counts to binary. These nouns are also processed using Lemmatizer from NLTK to greatly reduce the feature space.

2.2 Mapping image features to description

In the last section, we described how we used tags as output to match descriptions. One key take away from experimenting with the previous section is that when mapping something of a higher dimension to something with a lower dimension (80), a lot of information is lost. Therefore, we decided to use image features to capture the information that is lost in the tag space.

2.2.1 First attempt - Bilinear Pairwise Regression

Define $BoW_{description}$ as X, feature from fc1000/pool layer as Y, we sample 200000 examples and for pairs that are from the same image we set its output Z to 1 and -1 otherwise. Putting this in mathematical form, we formulated the following function, where i is the index of X and j is the index of Y.

$$Row\_Sum(XW * Y) = \begin{cases} 1 & \text{i = j} \\ -1 & \text{i} \neq \text{j} \end{cases}$$

We want to minimize the loss function, here we used hinge loss which is popular for image classification tasks

$$L_{\text{hinge}} = \max(0, 1 - z * Row\_Sum(XW * Y))$$

The model was built using tensorflow. The output of this model directly correspond to scores given images.

2.2.2 Partial Least Square Regression

After some literature review, we found that the better way of relating one source of data to another in a cross-modal setting is doing Partial Least Square (PLS). PLS will try to find the multidimensional direction in the X space that explains the maximum multidimensional variance direction in the Y space (projection on to latent structure). The idea is similar to PCA except that it models the relation between two variables. In PLS Regression (PLSR), it also maximizes fit and minimizes misfit while maximizing correlation between X and Y.

$$max \quad corr(X_k u, Y_k v) * std(X_k u) std(Y_k u)$$

such that $|u| = 1, |v| = 1$.

The end result of the regression is then used to calculate cosine similarity.

In the end, we trained 3 different PLSR models each hoping to extract different information from the input so that they can be ensembled together to form a robust model.

$$fc1000 -> PLSR(n\_components = 400) -> BoW_{description\_nouns\_binary}$$

$$pool -> PLSR(n\_components = 200) -> BoW_{description\_nouns\_adjectives\_verbs\_binary}$$
$$pool -> PLSR(n\_components = 200) -> BoW_{description\_nouns\_adjectives\_verbs\_binary}$$

The reasoning behind the 3 models is as follows: FC1000 contains object information so we used it to map image to nouns and nouns only in description. Pool layer contains comprehensive information about the images so we used it to map image to nouns adjectives and verbs in description. Finally, we trained the mapping of adjectives and verbs using the Pool layer hoping that this classifier can differentiate images with similar objects but disparate adjectives and verbs.

2.3 Ensemble model

With four different models at hand, we needed a way to blend them together in a way that boosts the performance of the combined model.

2.3.1 First attempt - Vanilla averaging

Our first attempt simply average the score of our models together to produce the top 20 results.

2.3.2 Second attempt - Weighted average

In the second attempt, we decided to do a linear weighting to the models. For 2 models this is simple to do.

$$score = \alpha * model1 + (1 - \alpha) * model2$$

To find the value of α, one can use grid search to find the optimal value. However, this process is tedious and slow and grows exponentially with the number of models used.

2.3.3 Final version

We wanted to see if there is a even more generalized way to do this. We also hypothesized that different models performs better on different inputs. Intuitively, we want to calculate the weights αi based on the input $BoW_{description}$. Formally, we want

$$M = \sum_{k=1}^{4} F(BoW_{description})_k * M^k.$$

We denote four cosine similarity matrices obtained by mentioned four models as M^k where $k = 1, 2, 3, 4$ in which m_{ij}^k is the similarity between query q_i and image v_j, and then we formulated an ensemble model to combine them to the eventual similarity matrix M. Recall that the BoW vectors for nouns, noun-adjetive-verb, and adjective-verb are respectively denoted as x_n, x_{nav}, and x_{av}, our ensemble model aims to learn $w = [w_1, w_2, w_3, w_4]^T$ that generate four coefficients

$$\begin{aligned} \alpha_1 &= w_1^T x_n \\ \alpha_2 &= w_2^T x_n \\ \alpha_3 &= w_3^T x_{nav} \\ \alpha_4 &= w_4^T x_{av}, \end{aligned} \tag{1}$$

such that they minimize the objective function given as

$$\underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{N} (\frac{1}{N} \sum_{j=1}^{N} m_{ij} - m_{ii}), \tag{2}$$

where m_{ij} is the element of M which is blended as

$$M = \sum_{k=1}^{4} \alpha_k M^k. \tag{3}$$

Model used	Accuracy on test set
Naive KNN	0.1586
Handcrafted codebook KNN	0.1864
SVM to tags	0.2261
Bilinear Pairwise Regression (BPR)	0.1907
Average of SVM to tags + BPR	0.2686
Ensembled SVM to tags + Bilinear Pairwise Regression	0.2885
Average of BPR, PLS_{nouns_fc1000}, SVM to tags	0.3381
Ensembled PLS_{nouns_fc1000} + SVM to tags	0.3517
Ensembled PLS_{nouns_fc1000} + PLS_{nav_pool} + SVM to tags	0.3871
Ensembled PLS_{nouns_fc1000} + PLS_{nav_pool} + PLS_{av_pool} + SVM to tags	0.4024

Table 1: Compared several experiments we submitted to Kaggle.

The intuition of this loss function is simple: as the diagonal element of the similarity matrix indicates that the query and the image match, the loss function that we defined as

$$l(m_{i:}) = \frac{1}{N}\sum_{j=1}^{N} m_{ij} - m_{ii} \tag{4}$$

favors to maximize the values gap between the matched item and others.

In predict phase, we chose the top 20 similar images based on the similarity value we obtained from ensembled model as the retrieval images to give back.

3 Experiment

We conducted multiple experiments to validate our proposed image retrieval models. We trained the models on the training set containing 10000 images and submit to Kaggle the predictions for test set that has 2000 images. The competition in Kaggle used the Mean Accuracy Precision (MAP) to evaluate model's performance. The MAP score is given by

$$s = \frac{20 + 1 - i}{20} \tag{5}$$

where i is the ground truth's rank if it is in the retrieval results.

3.1 Utilized open source software

numpy to manipulate arrays, matrices, and tensors.

nltk to pre-process descriptions, including tokenization, lemmatization, and stop word removal.

pandas to read structural data files.

scikit-learn common machine learning libraries

TensorFlow for writing regression algorithms with non standard loss function.

3.2 Results

Fig.2 demonstrates two examples of our retrieval results.

Please see table 1 for the complete list of different models used.

4 Conclusion

We have demonstrated the efficacy of using ensemble of PLSR models and SVM text mappings to achieve good results in image retrieval with text queries.

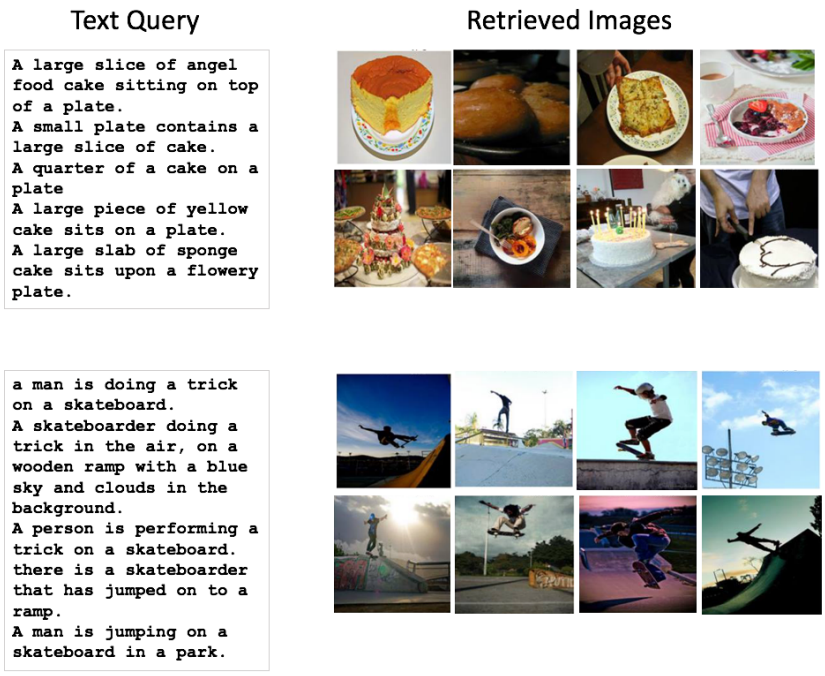

Figure 2: Examples of query results.

4.1 Future works

We believe that our model can be improved by adding more disparate models into the ensemble. Due to the time constraint, we did not have time to train more. Also, since PLSR is a linear regression method in the projection space, it is possible that using Neural Network to do a non linear regression from images feature to $BoW_{descriptions}$ can perform even better. Lastly, deep models combing text and images seems very promising based on literature review.

Acknowledgments

The authors would like to thank Serge Belongie, Yin Cui and Longqi Yang for their insightful advice.

References

[1] Wang, K., Yin, Q., Wang, W., Wu, S., & Wang, L. (2016). A Comprehensive Survey on Cross-modal Retrieval. arXiv preprint arXiv:1607.06215.

[2] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[J]. arXiv preprint arXiv:1512.03385, 2015.

[3] Pedronette, D. C. G., Almeida, J., & Torres, R. D. S. (2014). A scalable re-ranking method for content-based image retrieval. Information Sciences, 265, 91-104.

[4] Chen, Y., Wang, L., Wang, W., & Zhang, Z. (2012, September). Continuum regression for cross-modal multimedia retrieval. In 2012 19th IEEE International Conference on Image Processing (pp. 1949-1952). IEEE.

[5] Rasiwasia, N., Costa Pereira, J., Coviello, E., Doyle, G., Lanckriet, G. R., Levy, R., & Vasconcelos, N. (2010, October). A new approach to cross-modal multimedia retrieval. In Proceedings of the 18th ACM international conference on Multimedia (pp. 251-260). ACM.

[6] Li, Y., Yang, M., & Zhang, Z. (2016). Multi-View Representation Learning: A Survey from Shallow Methods to Deep Methods. arXiv preprint arXiv:1610.01206.

[7] Li, Y., Yang, M., & Zhang, Z. (2016). Multi-View Representation Learning: A Survey from Shallow Methods to Deep Methods. arXiv preprint arXiv:1610.01206.

[8] Sculley, D. (2010, July). Combined regression and ranking. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 979-988). ACM.

https://docs.scipy.org/doc/numpy/

http://scikit-learn.org/

https://pypi.python.org/pypi/Theano