A Cascade Ranking Model for Efficient Text to Image Retrieval

Anonymous Author(s)
Affiliation
Address
email

Abstract

The ubiquitous accessibility and explosive increase of image data has resulted in the property of research in image retrieval. In this paper, we address the task of text-based image retrieval, to select a set of images from a large image database. We vectorized the query descriptions to a sparse binary matrix, employed PLSRegression to learn the common subspace between the visual and textual features, made distance-based relevant image predictions, and rank the relevent images based on a combination of predicted l2-distance and category jaccard similarity. Experimental results demonstrate that our method effectively utilizes both visual and semantic meanings of the cross-modal information, outperforming other teams on Kaggle competition, and can exploit large scale vision and language datasets for knowledge transfer.

1 Introduction

The task of image retrieval means that, if given a specific description, retrieving all images relevant to that description within a potentially very large database of images. As multimedia data such as texts and images are growing exponentially on digital devices, it becomes increasingly difficult for users to search information effectively and efficiently. Therefore, cross-modal retrieval has attracted a considerable attention from many researchers. The current research effort is to design variant approaches to solve the cross-modal challenges more accurate and more scalable.

1.1 Problem Definition

As part of the Cornell Tech machine learning community, we are assigned with the task to build a search engine using novel statistical learning techniques with a goal to retrieve images that share the same semantics with the given quey description.

Data During training, we have a dataset of 10,000 image samples, each with the follwing data available for learning: a 224x224 JPG image, a list of tags indicating objects appeared in the image, a five-sentence description and feature vectors extracted using ResNet, a state-of-the-art Deep-learned CNN. The features are extracted from pool5 and fc1000 layer.

During testing, your system matches a single five-sentence description against a pool of 2,000 candidate samples from the test set. Each sample has: a 224x224 JPEG image, a list of tags for that image, ResNet feature vectors for that image.

Output For each description, our system ranks each testing image with the likelihood of that image matches the given sentence. Our system then returns the name of the top 20 relevant images.

Evaluation Metric We need to rank-order test images for each test description and the rating system uses MAP@20 as the evaluation metric. If the corresponding image of a description is among our

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

algorithm's 20 highest scoring images, this metric would then give a certain score based on the ranking of the corresponding image.

2 Model Overview

In this section, we describe our model for text to image retrieval and the training procedure in details. At test time, 2,000 images and 2,000 natural language textual descriptions are provided. For each description, the system needs to select from the 2,000 test images a subset of 20 target images that match the query text and sorts the candidates by relevancy ranking.

Figure 1 shows the architect of our final learning model. Before reaching the final model, we tried different approaches to analyze the cross-modal relationship between the image and the text. For each approach, we tried to utilize different features and different parameters for fine-tuning.

We divided the problem into the follwing steps: 1). extracting features from training descriptions, 2). finding correlations between text features and image features, 3). combining the components above to produce l2-distances for all 2,000 testing images, 4) re-ranking the images by combining l2-distance and jaccard similarity then retrieving the top 20 candidates for a given text query.

2.1 Extracting feaatures from descriptions

2.1.1 Attempt One: Doc2Vec and Word2Vec

Doc2Vec is an unsupervised algorithm that generate vector for paragprahs in documents. The feature vectors generated by Doc2Vec can be utilized to find similarity between documents. Upon given the data, we quickly realized that we need to identify similar descriptions to perform the task given. Yet it performs poorly. Our observation shows that although Doc2Vec is able to classify similar documents, it cannot distinguish different nouns that captrues important meaning in the descriptions. For example, Doc2Vec would give a similar score to "A man walks down the street" and "A woman walks down the street". Therefore, it is not ideal for the given task.

We then turned to Word2Vec, which takes an average of the sum of the word vectors to generate a sentence vector. The performance is not that ideal either. We suspect the reason is that the training sample is too small. For example, Google GloVe is an unsupervised learning algorithm that performs exceptionally for obtaining vector representations of words. Yet the training is performed on aggregated global word-word co-occurence statistics.

2.1.2 Attempt Two: TF-IDF

TF-IDF refers to "Term Frequency-Inverse Document Frequency", and the TF-IDF weight is used to evaluate how important a word is in a document. A term frequency(TF) is a count of how many times a word occurs in a given document(like bag of words), and the inverse document frequency(IDF) is the number of times a word occurs in a corpus of documents. Words that are used frequently in many documents will have a lower weighting while infrequent ones will have a higher weighting. The final weight is the product of TF and IDF, by calculating the final weight of given words in documents, we can form the vector for every description and then use the vector to compute the similarity.

Firstly, we notice that every description is in natural language. Therefore, we pre-process the descriptions by removing all punctuation, using NLTK module to remove all stopwords in the text, convert all letters in the descriptions into lower letter. After preprocessing, we processed descriptions corresponding to a certain image into Bag of Words, and we calculate the value vector for every description file. We then calculate the TF-IDF array for each tag file after doing similar preprocessing. By using vectors of descriptions and tags, we calculate cosine similarity between vectors. We ran a naive experiment to predict testing results and used it as a baseline to check our performance.

2.1.3 Attempt Three: SVM Tag Generation

We did some exploratory data analysis so that we found that there are 80 different tags for training set. Therefore, we trained 80 One vs. All SVM classifier with hinge loss and tried to predict if the predicted tags are in a given description. Experimental results shows that the following text feature

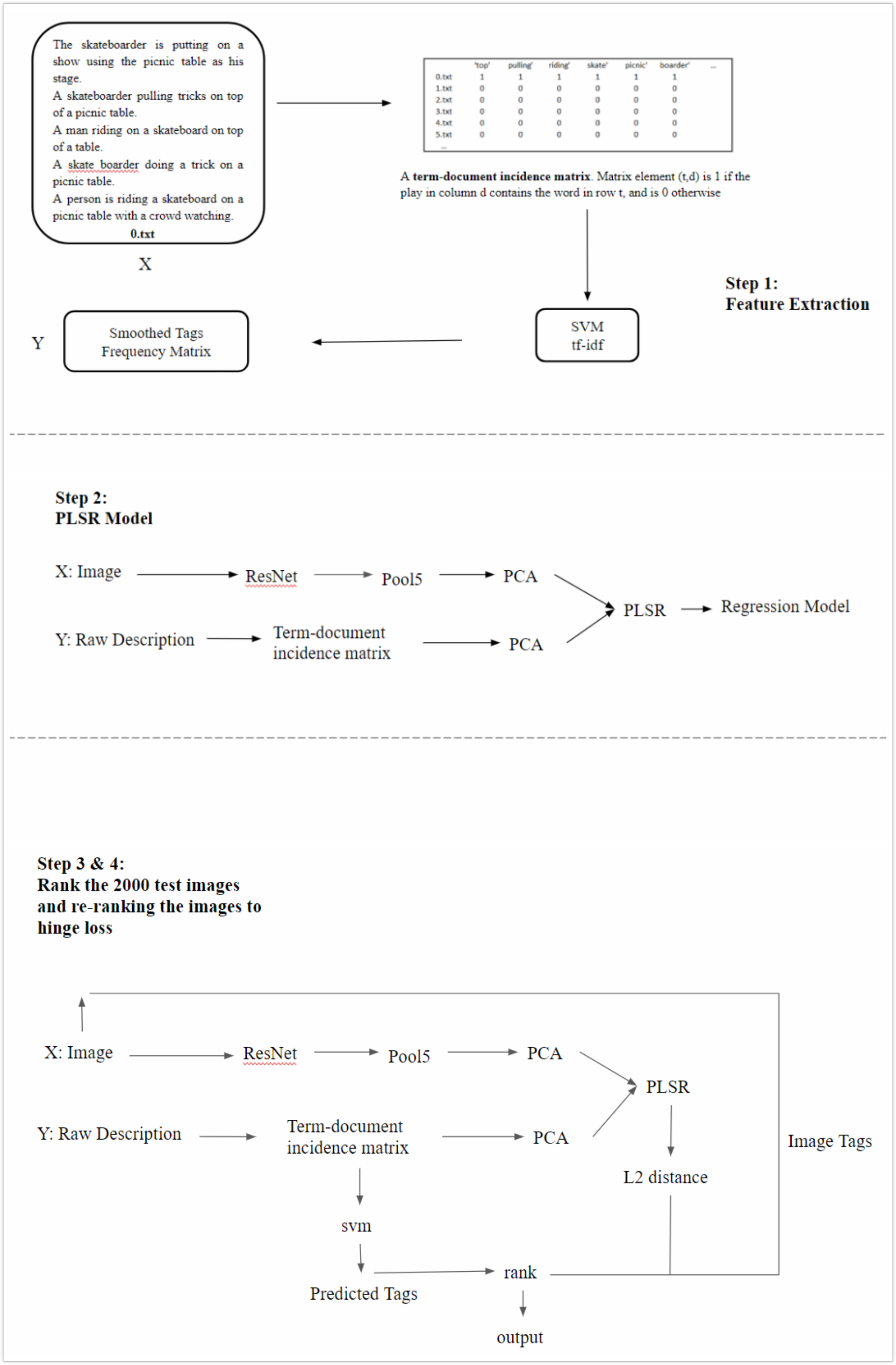

Figure 1: Model Architecture

extraction performs better than our SVM tag generation. Yet we were able to utilize these models generated to obtain results that subsequetly serves as input for our re-ranking algorithm in step 4.

2.1.4 Final Version: Term-document incidence matrix

Our final attempt is the simplest form of feature extraction for the text description. That is to create a term-document incidence matrix. First, we scan through all the documents to find a set of unique words after removing the stopwords that appears in the corpus. The set of unique words then becomes a feature vector for each query description. The matrix element is 1 if a given word appears in the description, and is 0 otherwise.

Instead of generating a sparse binary matrix, we have also tried recording the counts and thereby a document term matrix. Experimental result shows the incidence matrix works better in our case as the document term matrix would give an unfair weight to two documents if the overlapping term appears multiple times when comparing similarity. In reality, if two descriptions have many intersections yet the intersections are pointing to a single word, it is a weaker signal of similarity than if two descriptions have less intersections yet each intersection represents an unique word. We have also tried recording the counts of unique nouns that appears in the corpus. Yet experimental result shows that incidence matrix with all the words except stop words works the best with our model.

We then utilized the term document incidence matrix as our textual features to map the relationship with the visual features. Detail of the model is given below. For simplicity, we will refer to term document incidence matrix(TDIM) as text feature below.

2.2 Finding correlations between image features and text features

We have tried to use textual features alone, including text features from descriptions and tags features generated by our SVM model, to make predictions. It is forseeable that the accuracy would not be ideal. Since the texutal features are manually annotated, descriptions and tags usually do not consist a comprehensive information of the image. As a lot of information is lost in this way, it is necessary to incorporate the image features (fc1000 and pool5 from the ResNet) and construct a mapping between text features and images features.

2.2.1 Step One: Principal Component Decomposition

In large scale data mining and predictive modeling, especially for multivariate regression, we often start with a large number of possible explainatory/predictive variables. Therefore, variable selection and dimension reduction is a major task before modeling our cross-modal relationship.

As our text feature is represented by a sparse binary matrix, it makes sense to use principal component analysis to perform linear dimentionality reduction using Singular Value Decomposition of the data to project it to a lower dimenstional space. PCA is also a powerful tool for pattern recognition which is often used in image processing. Therefore, we perform reduction technique on both features before training our model that maps one to the other.

$$pool5 \rightarrow PCA(n-components = 1000) \rightarrow Reduced\_pool5$$

$$TDIM \rightarrow PCA(n-components = 3500) \rightarrow Reduced\_TDIM$$

2.2.2 Step Two: Partial Least Square Regression

After some literature review, we found some popular methods for establishing inter-modal relatioships between data from diffrent modalities, including Canonical Correlation Analysis, Bilinear Model, and Partial Least Squares. After many experimentations, we decided to use PLSRegression that linearly map images with different modalities to a common subspace. According to Chen et al., they apply PLS to image features into the text spaces in multimedia retrieval, then learn a semantic space for the measure of covariance between the two modalities. In PLSRegression, it maximizes correlation between the input and output as well as maximzes fit while minizing misfits.

We used different combination of X and Y to find a model that best explains the information presented by the given dataset. For Y value, we used all the textual features mentioned in Section 2.1. For X value, we conducted different trials with fc1000, pool5, and stacking both fc1000 and pool 5 as our input feature. After experimenting with different combinations, we identified that term document incidence matrix and pool5 features gives us the best accuracy with the model proposed.

$$pool5 \rightarrow PLSR(n - components = 800) \rightarrow Reduced\_TDIM$$

2.3 Calculating l2-distances for all 2000 images

After we trained the PLSRegression model with features and parameters listed above, we then used the end result of regression to calculate euclidean distance between each predicted result and the true description features across the 2,000 test images. Then we rank the 2,000 test images based on the l2 distance in an ascending order to obtain a preliminary ranking results for our test set descriptions.

2.4 Re-ranking the images

We then introduce another similarity measure to help us capture the categorical connection between query descriptions and images. The similarity is defined as follows:

$$sim = \frac{|D| \cap |I|}{|I|}$$

Where D is the set of predicted tags for query document and I is the set of tags for image candidates. Finally, for a give query document D we will reward pair-wise $l2$ distance from 2.3 and pick the top 20 as our output:

$$min_{l2}\{l2_{ij} - sim_{ij}\}$$

3 Experiments

We conducted multiple experiments to blend our features and models in a way that boosts our perforamce. After a number of trials, we obeserved that with the same model, using pool5 features as our X for PLSRegression performs much better than fc1000. This makes sense because the 1000-dimensional layer is the one just before the softmax, so the i dimesion captures how confident the network is that the image contains a class i. In other words, fc1000 provides categorical information of the image. In comparision, the pool5 layer captures some kind of semantic meaning, although each dimension is not necessarily interpreatable. In comparision, pool5 has more images features and may share the same semantics meaning with the text query. We suspects that might be why it outperforms fc1000 by a siginificant amount in our model setup.

As for our text features, we find the less processing, the better it works in terms of letting image features switch into its space in the subsequent regression analysis. The resulted common semantic space provides a good measure of correlation between the two different modalities.

3.1 Experimental Results

Figure 2 displays a list of testing accuracy from our selected testing trials. Our best result achieved on the private leader board is 0.42983.

4 Conclusion

In this work, we demonstrated a two stage text-based image retrieval with a re-ranking method that significantly boosts the performance of our retrieval task (see Figure 3). Cross-modal retrieval is a more powerful and effective way to multimodal retrieval than the traditional single modality based techniques. In the proposed model, we levaraged the text annotations to learn a visual embedding of the images and showed that this embedding predicts very well when given a text query (see Figure 4 and 5). In addition to locate the corresponding image, our model is also able to group images that share the same semantic meanings. Finally, a bottleneck of our performance is due to the inconsistency in manual annotations, as there are 84 training images that lack of tag descriptions. Therefore, our joint textual and visual model can leverage refining dataset to provide a more accurate method to query large image database.

Model Used	Accuracy on Test Set
TF − IDF	0.14795
SVM tags	0.15024
Word2Vec	0.16075
$PLSR_{W2V\_fc1000}$	0.16430
$PLSR_{TDIM_{NOUNS} \rightarrow fc1000}$	0.25095
$PLSR_{TDIM_{NOUNS} \rightarrow pool5}$	0.32139
$PLSR_{TDIM \rightarrow pool5}$	0.34639
$PLSR_{pca+pool5+svm+naive(euc-jaccard/\min\_dist)}$	0.35441
$PLSR_{pool5+svm+naive(euc-jaccard(len(test\_img))/\min\_dist)}$	0.38831
$PLSR_{pool5+svm+naive(euc-jaccard(len(test\_img))/\min\_dist)}$	0.42983

**The disparity between the last two entries is due to the difference of n components。

**The lower one is obtained by using n components = 800

**The higher score is obtained by using n components = 1000

Figure 2: A selection of our submitted Kaggle results

Text Query

A sharp bladed knife left out on the ground.
A large knife is on the ground in pine needles.
A single-edged knife with a black handle lying on the ground.
A knife with a curved silver blade and black handle resting on ground covered with debris.
Whoever owns this deadly knife means serious business.

Re-ranked Image

Index 26

After re-ranking

Index 1

Figure 3: Sample Re-ranking Performance

4.1 Future Works

Due to the limited amount of time and computation resources we have, we can only run a small number of experiments before the deadline. We believe that we can improve our model by leveraging the fc1000 image features and combining it with the existing model to build an ensemble model. Lastly, non-linear modeling might be an area worth exploring, as literature review shows that methods such as Deep Canonical Correlation Analysis(DCCA) and Multimodal Latent Binary Embedding(MLBE) seems promising.

Acknowledgments

The authors would like to thank Serge Belongie for the fantastic lectures. The authors would also like to thank Michael Wilber and Xun Huang for the insightful replies on Slack.

A pack of zebra walking along a grassy field.
several wild zebra's standing in a field of grasses and shrubs
A bunch of zebras standing in a field
A pack of zebras stand on the plain looking into the distance
A herd of zebras are standing on a grassy field.

Text Query

Retrieved Images

Figure 4: Sample Query Result

A man in a baseball uniform kneeling down while holding a baseball bat.
This ball player is posing for a picture in his Ranger's uniform.
A baseball player wears a Rangers outfit while kneeling and holding a bat.
A baseball player poses for a picture with his baseball bat.
Baseball player posing in his uniform holding a baseball bat.

Text Query

Retrieved Images

Figure 5: Sample Query Result

References

[1] Albert Gordo & DianeLarlus (2017) *Beyond instance-level image retrieval: Leveraging captions to learn a global visual representation for semantic retrieval*

[2] Christopher D. Manning & Prabhakar Raghavan & Hinrich Schütze (2008) An example information retrieval problem *Introduction to Information Retrieval*

[3] Wengang Zhou & Houqiang Li & Qi Tian (2017) *Recent Advance in Content-based Image Retrieval: A Literature Survey*

[4] ShangWen Li & Sanjay Purushotham & Chen Chen & Yuhuo Ren (2017) *Measuring and Predicting Tag Importance for Image Retrieval*

[5] Jeffrey Pennington & Richard Socher & Christopher D. Manning *GloVe: Global Vectors for Word Representation*

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

http://scikit-learn.org/

https://docs.python.org/3/library/pickle.html

https://pypi.python.org/pypi/nltk