CS5785 Fall 2018 Final Report

December 10, 2018

Abstract

Our final project is to build a large-scale image search engine, which can search for relevant images given a natural language description. We were given a training set of 10,000 samples and a test set of 2,000 samples, each with a set of descriptions, a 224×224 JPG image, a list of tags indicating objects appeared in the image and ResNet features. Our team tried different approaches and generated different results. The best performing algorithm received an accuracy of 41.76% on Kaggle. This algorithm leveraged Ridge Regression.

1 Introduction

The job here is for each 5-sentences description, our system must rank-score each testing image with the likelihood of that image matches the given sentences. The system then returns the name of the top 20 relevant images.

Our approaches first use some description based algorithm to prepossess input descriptions, and then train a model with tags or features as outputs. Once this is achieved, we then find the nearest neighbours in the test set of images based on distance measurement.

This report describes different approaches we used to map test descriptions to some estimated metric, like tags or ResNet features, and then map to test images that are most related to the descriptions.

2 Experiment

2.1 Algorithms

In this project, we used 6 different algorithms, that is Bag of Words, Word2Vec, Random Forest, Ridge Regression, PLS Regression and kNN.

2.1.1 Bag of Words

A problem with modeling description is that it is messy, and techniques like machine learning algorithms prefer well defined inputs. Machine learning algorithms cannot work with raw description directly; the description must be converted into vectors of numbers first.

Bag of Words model is a way of extracting features from description for use in modeling. It is called a “bag” of words, because any information about the order or structure of words in is discarded. The model only concern of whether known words occur in the document, and a measure of presence of known words, not where it appear in the document.

So we dropped the stopwords, numbers and punctuation in description, focus on content alone. Instead of recording a binary number of whether the word exists or not, we calculated TF-IDF of each word in the bag of words for all descriptions.

TF-IDF

TF-IDF stands for term frequency-inverse document frequency, it is often used in mining text and comparing word documents. TF-IDF is intended to reflect how relevant a term is in a given document. A word’s relevance is proportional to the amount of information it gives regarding the context.

The intuition behind it is that if a word occurs multiple times in a document, we should increase its relevance as it should be more meaningful than other words that appear fewer times (TF). Meanwhile, if a word occurs many times in a document but also along many other documents, maybe it is because this word is just a frequent word, not because it is more relevant (IDF).

In our case, we calculated TF-IDF and then made bag of words both for descriptions and image tags.

2.1.2 Word2Vec

Word2Vec model is used for learning vector representations of words, also called ”word embedding”. It takes as its input a large corpus of words and produces a vector space, typically of several hundred dimensions. Each unique word in the corpus is assigned with a corresponding vector in the space. Words that share common contexts in the corpus are located in close proximity to one another in the vector space. The model also captures semantic information about words and their relationships to one another.

In our case, we used Word2Vec model on both descriptions and image tags to map text into vector space.

2.1.3 Ridge Regression

Ridge Regression is a method in linear regression, it is a technique for analyzing multiple regression data that suffer from existence of near-linear relationships among the independent variables. When this occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value. By adding a degree of penalty to the regression estimates, ridge regression reduces the standard errors, which will give estimates that are more reliable.

In our case, we used descriptions as input for Ridge Regression, as for output, we tried ResNet features and image tags.

2.1.4 Random Forest

Random Forest is consisted of a set of decision trees. Each tree is developed from a bootstrap sample from the training data set. When developing individual trees, an arbitrary subset of attributes is drawn, from which the best attribute for the split is selected. The final model is based on the majority vote from individually developed trees in the forest.

Unlike linear models, random forests are able to capture non-linear interaction between the features and the labels.

In our case, we used descriptions as input for Random Forest, as for output, we tried ResNet features and image tags.

2.1.5 PLS Regression

PLS Regression combines principal component analysis and multiple regression together.It reduces the features to a smaller set of uncorrelated components and performs least squares regression on these components, instead of on the original data sets. PLS regression is especially useful when there's more features than labels, and in this situation, ordinary least square regression is not helpful.

In our case, we used ResNet Features as input for PLS Regression, as for output, we used descriptions.

2.1.6 kNN (K Nearest Neighbours)

K Nearest Neighbours is a supervised method for classification and regression.

In our case, we used distance matrix to calculate the test distance, sorted the result and took the top 20 images based on the ranking.

2.1.7 Ensemble

Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.

2.2 Experiments on different combinations

2.2.1 Bag of Words for Descriptions to Tags

In this method, we built models with description data as features and tags as labels. To better build our models, we firstly constructed bag of word models for both descriptions and tags. Since we did not want binary numbers indicating whether each word exists or not, we calculated the TF-IDF for words in the bag of words model to get the weight of each word by considering the term frequencies and inverse document frequencies.

Experiment 1: Ridge Regression
Try different alpha values: 0.01, 0.5, 1.0, 1.5, 2.0

Try different distance calculation methods:
a. KNN: ((np.expand_dims(x1, 1) - np.expand_dims(x2, 0)) ** 2).sum(2) ** 0.5
b. Cosine distance

Experiment 2: Random Forest
Set all parameters in random forest as default

Try different distance calculation methods:
a. KNN: ((np.expand_dims(x1, 1) - np.expand_dims(x2, 0)) ** 2).sum(2) ** 0.5
b. Cosine distance

2.2.2 Bag of Words for Description to ResNet Feature Vectors

In this method, we also used bag of words model and TF-IDF to change the description to weight of each word; however, instead of tags, this time we used ResNet feature vectors including 1,000 dimensional features from classification layer (fc1000) as labels. Or, we can use intermediate ResNet feature Vectors including 2048 vectors as labels.

Experiment 3: Ridge Regression
Try different alpha values: 1.0, 10.0, 50.0, 100.0

Try different distance calculation methods:
a. KNN: ((np.expand_dims(x1, 1) - np.expand_dims(x2, 0)) ** 2).sum(2) ** 0.5
b. Cosine distance

Try ResNet feature vectors (1000) and ResNet intermediate feature vectors (2048).

Experiment 4: Random Forest
Set all parameters in random forest as default

Try different distance calculation methods:
a. KNN: ((np.expand_dims(x1, 1) - np.expand_dims(x2, 0)) ** 2).sum(2) ** 0.5
b. Cosine distance

Try ResNet feature vectors (1000) and ResNet intermediate feature vectors (2048).

2.2.3 Word2Vec for Description to Tags

In this method, we built models with description data as features and tags as labels. However, unlike the method in 2.2.1, we formed the description vectors by using word2vec, which provides a pre-trained 300-dimensional vector representation of most words in the English language. The feature vector for a given description was then formed by averaging the 300-dimensional word2vec vectors of all the words in the description.

Experiment 5: Ridge Regression
Try different alpha values: 0.01, 0.5, 1.0, 1.5, 2.0

Try different distance calculation methods:
a. KNN: ((np.expand_dims(x1, 1) - np.expand_dims(x2, 0)) ** 2).sum(2) ** 0.5
b. Cosine distance

Experiment 6: Random Forest
Set all parameters in random forest as default

Try different distance calculation methods:
a. KNN: ((np.expand_dims(x1, 1) - np.expand_dims(x2, 0)) ** 2).sum(2) ** 0.5
b. Cosine distance

2.2.4 Word2Vec for Description to ResNet Feature Vectors

In this method, we still used word2vec to form the description vectors; however, instead of tags, this time we used ResNet feature vectors including 1,000 dimensional features from classification layer (fc1000) as labels. Or, we can use intermediate ResNet feature Vectors including 2048 vectors as labels.

Experiment 7: Ridge Regression
Try different alpha values: 0.01, 0.5, 1.0, 1.5, 2.0

Try different distance calculation methods:
a. KNN: ((np.expand_dims(x1, 1) - np.expand_dims(x2, 0)) ** 2).sum(2) ** 0.5
b. Cosine distance

Try ResNet feature vectors (1000) and ResNet intermediate feature vectors (2048).

Experiment 8: Random Forest
Set all parameters in random forest as default

Try different distance calculation methods:
a. KNN: ((np.expand_dims(x1, 1) - np.expand_dims(x2, 0)) ** 2).sum(2) ** 0.5
b. Cosine distance

Try ResNet feature vectors (1000) and ResNet intermediate feature vectors (2048).

2.2.5 Bag of Words for ResNet Feature Vectors to Description

In this method, we also used bag of words model to form the description vectors; however, this time we used ResNet feature vectors including 1,000 dimensional features from classification layer (fc1000) as features and descriptions as labels. Or, we can use intermediate ResNet feature Vectors including 2048 vectors as features.

Experiment 9: PLS Regression
Try different n_components values: 2.0, 10.0, 50.0, 100.0, 200.0

Try different distance calculation methods:
a. KNN: ((np.expand_dims(x1, 1) - np.expand_dims(x2, 0)) ** 2).sum(2) ** 0.5
b. Cosine distance

Try ResNet feature vectors (1000) and ResNet intermediate feature vectors (2048).

2.2.6 Bag of Words for Tags to Description

In this method, we also used bag of words model to form the description vectors and image tag vectors; and this time we used image tags as features and descriptions as labels.

Experiment 10: PLS Regression
Try different n_components values: 2.0, 10.0, 50.0, 100.0

Try different distance calculation methods:
a. KNN: ((np.expand_dims(x1, 1) - np.expand_dims(x2, 0)) ** 2).sum(2) ** 0.5
b. Cosine distance

3 Result

Method	Paramters	MAP@20 Score
(1)Desc(BoW) to Tags, Ridge	Alpha = 1	0.24
(2)Desc(BoW) to Tags, RF	N/A	0.15
(3)Desc(BoW) to ResNet Feature, Ridge	Alpha = 10	0.29
(4)Desc(BoW) to ResNet Feature, RF	N/A	0.10
(5)Desc(word2vec) to Tags, Ridge	Alpha = 1	0.01
(6)Desc(word2vec) to Tags, RF	N/A	0.01
(7)Desc(word2vec) to ResNet Feature, Ridge	Alpha = 1	0.13
(8)Desc(word2vec) to ResNet Feature, RF	N/A	0.07
(9)ResNet Feature to Desc(BoW), PLS	n_components=200	0.36
(10)Tags to Desc(BoW), PLS	n_components=100	0.27
Ensemble (3) and (9) and (10)	N/A	0.42

Desc: Description
Ridge: Ridge Regression
RF: Random Forest
PLS: PLS Regression

From the results, we found that the intermediate ResNet feature vectors to description with bag of words model, using PLS regression and cosine distance works the best. While the second best is image tags to description with bag of words model, using PLS regression and cosine distance. As for the third best, it is description with bag of words model to intermediate ResNet feature vectors, using ridge regression and cosine distance. We ultimately used an ensemble method to integrate this three algorithms and the score reached 0.42.

We also observed that bag of words out performed word2vec, both PLS Regression and Ridge Regression performed better than Random Forest, and cosine distance performed better than KNN with the distance formula above. In addition, when we use the ResNet feature vectors, the intermediate one worked better.

4 Discussion

There are many strategies to find the relationship between input descriptions and the top-match images. We used different methods, such as word2vec, bag of word and TF-IDF to deal with descriptions and image tags.We also used different models to fit the input and output data. We did experiment with random forest, ridge regression, PLS regression and used cross-validation to try different parameters and search for the one with highest score. Different models always generate scores with big difference, while change parameters results in small change in term of score. We also used different data sets as input and output, including image tags, two types of ResNet features and description.

We did many experiments and the results were not always satisfying. Sometimes we can make progress based on some small change in the code, like the method we used to calculate distance matrix, but sometimes we changed the model in the expectation that the score would have a great improvement and it turns out that it failed. Generally speaking, we still made progress through out the whole process although it's not a linear improvement.

Lastly, we found that the ensemble method with different models that were not correlated always gave better result than any model that involved.

5 References

[1]Opitz, D.; Maclin, R. (1999). "Popular ensemble methods: An empirical study". Journal of Artificial Intelligence Research. 11: 169&ndash, 198. doi:10.1613/jair.614.

[2]A Gentle Introduction to the Bag-of-Words Model https://machinelearningmastery.com/gentle-introduction-bag-words-model/

[3]Word2Vec Explained https://israelg99.github.io/2017-03-23-Word2Vec-Explained/

[4]Ridge Regression https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Ridge_Regression.pdf

[5]Random Forest https://docs.orange.biolab.si/3/visual-programming/widgets/model/randomforest.html

[6]What is partial least squares regression? https://support.minitab.com/en-us/minitab/18/help-and-how-to/modeling-statistics/regression/supporting-topics/partial-least-squares-regression/what-is-partial-least-squares-regression/

[7]https://scikit-learn.org/

[8]https://www.nltk.org/