Retrieving Relevant and Diverse Image from Social Media Images

Xi Chen
Peking University
Beijing, China
mrcx@pku.edu.cn

Haokun Liu
Peking University
Beijing, China
haokun-liu@pku.edu.cn

Zhihong Deng
Peking University
Beijing, China
zhdeng@cis.pku.edu.cn

Yunlun Yang
Peking University
Beijing, China
incomparable-lun@pku.edu.cn

ABSTRACT

We describe our approach and its result for MediaEval 2015 Retrieving Diverse Social Images Task. The basic idea is removing the irrelevant images and then obtaining the diverse image using a greedy strategy. Experiment results show our method can retrieve diverse images with a moderate relevance to the topic.

1. INTRODUCTION

Retrieving social media information is currently a hot topic in research. The Retrieving Diverse Social Images task puts forward a problem on how to generate a brief summary up to 50 images which is both relevant and diverse from Flickr photos [1].

The problem is challenging. On the one hand, we need to filter out the irrelevant images in Flickr to guarantee the relevance. On the other hand, we need to reserve different images in order to enhance the diversity.

Many work on this problem have been done previous. Concetto and Simone proposed a method including cluster wise filtering which attained a high relevancy but lower the diversity [2]. Dang-Nguyen, Piras and Giacinto introduced a method removing irrelevant image before clustering, yield high score in both criteria [3], but this method may fail when irrelevant images is in the majority of the images. There are methods trying to make use of the given ground truth with a neural network [4], but not very effective in improving the diversity.

We propose a method considering both relevance and diversity, which can be describe briefly as extracting images from the given images through clustering and sorting with the help of the images from Wikipedia. The basic idea is removing the irrelevant images and then obtaining the diverse images using a greedy strategy.

2. APPROACH

Our approach contains 6 steps (Figure 1): pre-filtering, clustering, cluster-filtering, sorting, distributing and extracting. Each of them is described in the following.

2.1 Pre-Filtering

In this step, we filter out part of irrelevant images based on the following rules:

1. If the distance between spots for photography and the query is over 30km.
2. If the proportion of human faces in the images is over 0.05.

Because the face detector we used is not absolutely accurate, and some images including people are also relevant to the queries' topic, the images we filter out are not all correct but of a considerable accuracy.

MediaEval 2015 Workshop, Sept. 14-15, 2015, Wurzen, Germany.

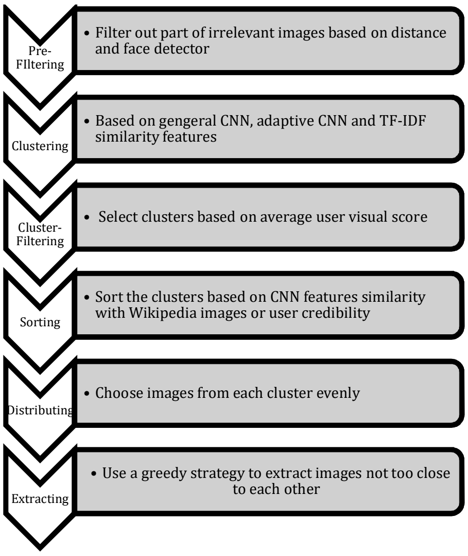
Pre-FIltering
• Filter out part of irrelevant images based on distance and face detector
Clustering
• Based on gengeral CNN, adaptive CNN and TF-IDF similarity features
Cluster-Filtering
• Select clusters based on average user visual score
Sorting
• Sort the clusters based on CNN features similarity with Wikipedia images or user credibility
Distributing
• Choose images from each cluster evenly
Extracting
• Use a greedy strategy to extract images not too close to each other

Figure 1. Schema of our approach

2.2 Clustering

In this step, we use general CNN(Convolution Neural Network) and adaptive CNN features, combined with cosine similarity of the image text TFIDF between the image and all provided images, to form a 4096+4096+N dimension feature, where N is the number of provided images after pre-filtering.

We use a hierarchical clustering algorithm [5] to cluster the image into 30 clusters.

2.3 Cluster-Filtering

In the previous step, similar images are clustered into the same cluster. As a result, in some cluster, most image are relevant, while in other clusters, most images are irrelevant. However, relevant images with different classes may not be directly clustered into different clusters. Instead, similar classes are probably clustered into one cluster.

We expect to remove irrelevant clusters first, and to deal with the diversity problem later. Using user information, we calculate the average user score of each cluster, and keep only the clusters whose score is higher than the average score of the whole image set and filter out the other.

2.4 Sorting

To sorting the clusters, we calculate the distance between each image in a cluster and the corresponding nearest image in Wikipedia, and sort the clusters based on the average distance.

	Visual features	Text features	User credibility
Run 1	CNN adaptive, CNN general	-	-
Run 2	-	TF-IDF	-
Run 3	CNN adaptive, CNN general	TF-IDF	-
Run 4	-	-	Visual score
Run 5	CNN adaptive, CNN general	TF-IDF	Visual score

Table 1. Features used in 5 runs

2.5 Distributing

After sorting, we need to determine the number of images to be extracted from each cluster i.e., distributing the number of images among clusters. Here we use uniform distribution i.e., choosing same number of images from each cluster. And if it can't be distributed evenly, redundant part will be chosen from the former clusters.

2.6 Extracting

After the previous distributing step, we apply a greedy strategy to extract images from each cluster. In the process of extracting, we choose the image in the first cluster with highest visual score to be the first image. And after that, in each step, image with highest score which is not among the top 15 nearest of any previously chosen image will be selected.

3. RESULTS AND DISCUSSION

3.1 Running Options

We submitted 5 runs as following:

Run 1. We use only image features in clustering and extracting, and clusters are sorted by size without being filtered. We distribute evenly and extract using a greedy strategy.

Run 2. We use only text features in clustering and extracting, and clusters are sorted by size without being filtered. We distribute evenly and extract using a greedy strategy.

Run 3. We use both image and text features. Clusters are sorted by size without being filtered. We distribute evenly and extract using a greedy strategy. Featured were weighted differently in clustering and extracting.

Run 4. We simply select the first fifty pictures sorted by user score.

Run 5. We use both image and text features. The clusters with low average visual score are filtered before sorted by distance to wiki image (sorted by average visual score if wiki is not available). We distribute evenly and extract using a greedy strategy. Featured were weighted differently in clustering and extracting.

Different features used are shown in Table 1 above.

3.2 Results

This section presents the experimental results achieved on development set (153 queries, 45,375 photos) and test set (139 queries, 41,394 photos). The CR@20 is number of covered classes among the top 20 results, P@20 is number of relevant photos among the top 20 results, and F1@20 is the harmonic mean of the previous two.

We obtained the best result at Run 5 in development set with F1@20 values of 0.5866, P@20 values of 0.7995, and CR@20 values of 0.4634. In the test for Single-topic, Run 5 gains the highest score on both relevance and diversity, while in the test for Multi-topic, though P@20 in Run 5 is not the highest, it's close to Run 4. While CR@20, F1@20 in Run 5 are lower than Run 3. Generally, according to the average performance, Run 5 get the best score on P@20 of 0.6957, CR@20 of 0.4435 and F1@20 of 0.5241.

	P@20	CR@20	F1@20
Run 1	0.6840	0.4270	0.5197
Run 2	0.6678	0.4068	0.4988
Run 3	0.6831	0.4347	0.5264
Run 4	0.7564	0.3159	0.4373
Run 5	**0.7995**	**0.4634**	**0.5866**

Table 2(a). Run performance on Development Set

	P@20	CR@20	F1@20
Run 1	0.6275	0.4053	0.4828
Run 2	0.6254	0.3980	0.4797
Run 3	0.6217	0.4086	0.4811
Run 4	0.6304	0.3245	0.4097
Run 5	**0.6804**	**0.4521**	**0.5298**

Table 2(b). Run performance on Test Set Single-topic

	P@20	CR@20	F1@20
Run 1	0.6864	0.4253	0.5090
Run 2	0.6964	0.4419	0.5261
Run 3	0.6850	**0.4626**	**0.5393**
Run 4	**0.7136**	0.3143	0.4109
Run 5	0.7107	0.4350	0.5185

Table 2(c). Run performance on Test Set Multi-topic

	P@20	CR@20	F1@20
Run 1	0.6572	0.4154	0.4960
Run 2	0.6612	0.4201	0.5031
Run 3	0.6536	0.4358	0.5104
Run 4	0.6273	0.3139	0.4103
Run 5	**0.6957**	**0.4435**	**0.5241**

Table 2(d). Run performance on Test Set

4. Discussion

In Development Set, our approach has achieved good results, while in Test Set, P@20 performs relatively poorly. We believe it is due to the method we used and problems we observed.

We noticed two problems in the experiments. One is that images in different classes may be clustered into the same cluster. So we utilize a greedy strategy, and set different image/test weight in the process of classification and extraction, which make a significant improvement in the experiments.

The second issue is that we found it a common scenario that users take multiple shots at one scene, causing irrelative images may be well clustered into several clusters with few relevant images. Because of the difficulties in removing irrelative clusters, we use user visual score to decide whether a cluster is irrelative or not. This method relies on the accuracy of user information. In the experiments it generally makes improvements, but it reduces the score at some test points, moreover, it makes the approach more sensitive to the quality of user information. Thus this problem has not yet reached a satisfactory solution.

5. CONCLUSION

We proposed an approach on how to generate a brief image summary considering both relevance and diversity from Flickr. This approach performs well as shown by our experiments. And we also put forward two problems which restrict the performance in the experiments.

Further research will be mainly on how to identify whether a similar picture is related to the topic or not.

ACKNOWLEDGMENT

This work is support by the Peking University Education Foundation (URTP2015PKU003).

6. REFERENCES

[1] Ionescu, B., Gînscă, A.L., Boteanu, B., Popescu, A., Lupu, M. ,Müller, H., Retrieving Diverse Social Images at MediaEval 2015: Challenge, Dataset and Evaluation. Working Notes Proceedings of the MediaEval 2015 Workshop, Wurzen, Germany, September 14-15, CEUR-WS.org, 2015;

[2] PeRCeiVe Lab@UNICT at MediaEval 2014 Diverse Images: Random Forests for Diversity-based Clustering Concetto Spampinato, Simone Palazzo

[3] Duc-Tien Dang-Nguyen, Luca Piras, Giorgio Giacinto, Giulia Boato, Francesco De Natale: Retrieval of Diverse Images by Pre-filtering and Hierarchical Clustering

[4] MIRACL's participation at MediaEval 2014 Retrieving Diverse Social Images Task: Hanen Karamti, Mohamed Tmar, Faiez Gargouri

[5] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.