UMONS @ MediaEval 2017: Diverse Social Images Retrieval

Omar Seddati, Nada Ben Lhachemi, Stéphane Dupont,
Saïd Mahmoudi
Mons University, Belgium
{omar.seddati,nada.ben-lhachemi,stephane.dupont,said.mahmoudi}@umons.ac.be

ABSTRACT

This paper presents the results achieved during our participation at the MediaEval 2017 Retrieving Diverse Social Images Task. The proposed unsupervised multimodal approach exploits visual and textual information in a fashion that prioritizes both relevance and diversification. As features, we used a modified version of the RMAC (Regional Maximum Activation of Convolutions) descriptor for visual information and word2vec-based weighted averaging for textual information. In order to provide an adaptive unsupervised solution, we combine these features with the DBSCAN (density-based spatial clustering of applications with noise) clustering algorithm. Our system achieved promising results and reached an F1@20 of 0.6554.

1 INTRODUCTION

Over the past decades, available image collections have seen consistent growth thanks to the easily accessible devices that we now use on a daily basis. These huge multimedia collections motivated researchers to look for efficient approaches for image retrieval. However, most of the approaches in this field primarily aim at the improvement of the relevance of the results, commonly neglecting the diversity aspect. The goal of the Retrieving Diverse Social Images Task [14] is to encourage researchers to propose new solutions that offer a good relevance-diversity balance. Participants are provided with several queries and up to 300 results corresponding to each query retrieved using the Flicker search engine. Each participating system is expected to provide a list with up to 50 ranked images per query that are both relevant and diversified. In addition to the images and the Flicker ranking, several metadata are provided such as username, credibility, etc. Both, visual information and metadata have been exploited in several ways by the participants of previous editions of the task [2, 11, 13]. The most used text-based features are Term Frequency-Inverse Document Frequency (TF-IDF)[9], Latent Dirichlet Allocation (LDA)[1], and word embeddings like word2vec [12]. For visual information, the most used features are Convolutional Neural Networks (CNN) based features. Several clustering algorithms have been explored such as k-means [13], X-means [13], agglomerative hierarchical clustering (AHC) [11], etc. In our work, we use word2vec-based weighted average as text-based features, an improvement of the RMAC descriptor [10] based on CNN features for visual information, and DBSCAN [4] as clustering algorithm.

MediaEval'17, 13-15 September 2017, Dublin, Ireland

2 APPROACH

In this work, we combine visual and/or textual descriptors with the DBSCAN algorithm at two different stages. In the first stage, we re-rank the provided list of results in order to remove some irrelevant images, while during the second stage, we aim to improve diversity. In our approach, the visual features based on the work of Tolias et al. [10]. Tolias et al. discarded the fully connected layers of a pre-trained CNN (VGG16) and used the resulting fully convolutional CNN for feature extraction. Let assume we have an input image I of size $(W_I \times H_I)$, the output feature maps (FMs) will form a $3D$ tensor in the form $C \times W \times H$ (where C is the number of channels, (W, H) the width and height of FMs). If we represent this $3D$ tensor as a set of $2D$ feature maps $\mathcal{X} = \{\mathcal{X}_c\}$, $c = 1...C$, we can compute the MAC (Maximum Activations of Convolutions) using the following equation:

$$f = [f_1 ... f_c ... f_C], \text{ with } f_c = \max_{x \in \mathcal{X}_c} x \tag{1}$$

In order to compute the RMAC descriptor, Tolias et al. proposed a simple approach to sample $R = \{R_i\}$, a set of square regions within $\mathcal{X}$, and compute the MAC for each region The sum aggregation of the resulting vectors after an l_2-normalization provides the RMAC descriptor (for more details please refer to the original paper [10]). In [5], Gordo et al. proposed two simple modifications to bring significant improvements to the RMAC representation: 1) using ResNet101 instead of VGG16; 2) three resolutions of the input image are feeded to the network. The RMAC descriptors are computed separately and l_2-normalized. Then, the three vectors are summed and l_2-normalized. In this work, we use the ResNet50 [6] and the publicly available Torch toolbox [3] to extract the RMAC descriptor with multi-resolution. However, instead of computing the RMAC descriptor separately for each resolution, we rescale the output feature maps of the three resolutions to the same resolution (the highest resolution) and sum them. Following, we compute the RMAC descriptor and do the sum-aggregation followed by an l_2-normalization (more information on the approach can be found in [7]). The RMAC descriptor has the advantages of keeping the aspect ratio of the inputs and encoding efficiently spatial information while keeping the size of the descriptor independent of the resolution of the input (but rather on the number of channels of the selected layer for feature extraction, which can be used as a parameter of the method).

3 EXPERIMENTAL RESULTS

In this section, we detail the five runs submitted by our team. Then, we briefly present the results obtained with the proposed approach on the development and test set.

Run 1: In the first run, only visual features are allowed to be used. Since the query is a textual query, we used the Flicker initial ranking and we made the assumption that the first three results (top

3) are relevant and can be used to generate a visual representation of the query. In order to re-rank the initial list, we extract the RMAC features from each image using three different input scales (where S is the largest side of the input and $S \in [550, 800, 1050]$). Then, we do a first clustering using the DBSCAN algorithm and we follow the following steps: 1. For each cluster, we find the closest feature vector to the cluster's center (V_{cl}); 2. We select the n clusters that contain the top 3 images ($n \leq 3$); 3. We compute the distance between each of the n clusters (centers) and the remaining r clusters, for each of these r clusters we keep as representative distance the minimal distance to one of the n clusters and we use it to re-rank the list of results; 4. We remove clusters that are at the bottom of the list, but we make sure that we keep enough clusters to have at least 150 images. This first stage enables us to remove some irrelevant images. In the second stage, we do another clustering (DBSCAN) and we sort the different clusters using the initial Flicker rank of the centroid. Then, we select one image per cluster until we obtain the required number of result images, if the last cluster is reached, we start again from beginning. Finally, we group the images that belong to the same cluster and present the results in the clusters order (based on the rank of centroids).

Note: *In order to correctly use the DBSCAN algorithm, we should carefully define the maximum radius ϵ. In our case, for each query, we compute a vector with n elements, where n is the number of available results and each element $e_i, i \in 1, ..., n$ is the minimal distance between image i and any other image. Finally, we use the median of this vector as ϵ, one as the number of minimum points, and the Manhattan distance as metric.*

Run 2: The second run uses the provided word2vec (dimensionality = 300) semantic vectors for English terms (trained over Wikipedia). Unlike TF-IDF or LDA, word2vec vectors do not look at word co-occurrence patterns but they have the advantage of addressing various sorts of similarities between words (syntactic and semantic). In order to select the textual information to use, we examined the devset queries and noticed that tags are more significant syntactically and semantically than other textual fields (e.g. title and descriptions). For each image, we compute the weighted average vector representation (as described in [8]) based on its tags . Then, we do clustering using The DBSCAN algorithm and sort the clusters using the distance between the query representation and the representation of the centroid of a given cluster. Finally, we re-rank the images following the same approach as the last step of run 1.

Run 3 & 4: In the third run, we concatenate the RMAC feature vector of Run 1 with the textual feature vector of Run 2 and followed the different steps of Run 1. In addition to that, just after the second clustering, we group the images uploaded by the same user and make sure that when picking images for the final ranking, we choose images from the different user groups of a given cluster. In Run 4, we followed the same steps as in Run 3, but we used only the RMAC descriptor as feature vector and the username grouping technique.

Run 5: In the fifth run, we first remove stop words from the queries. Then, we use each query to retrieve 10 images using Google image engine. We extract the RMAC features from these images and use them as a visual representation of the query as in Run 1.

Table 1: Results on the development and test set.

	Devset			Testset		
Run	P@20	CR@20	F1@20	P@20	CR@20	F1@20
Run1	0.6327	0.409	0.4722	0,6780	0,5599	0,5789
Run2	0,5595	0,4148	0,4581	0,5702	0,5834	0.5521
Run3	0.6359	0.4222	0.4827	0,6643	0,5780	0,5886
Run4	0.6373	0.4196	0.4825	0,6690	0,5649	0,5809
Run5	**0.7386**	**0.4467**	**0.5253**	**0,8071**	**0,5856**	**0,6554**

Next, we follow the same steps to re-rank the Flicker list. Since Google image results match better the queries, we can expect better visual representations, which allows us to use more efficiently the RMAC descriptor. In addition to that, as in Run 3 & 4, we use the grouping by username approach to further improve diversity.

Note: *in order to retrieve enough results from Google image and enhance diversity, we used the query in the following way: let's assume that we have a query with five words w_1, w_2, w_3, w_4, w_5, we use the following for image crawling:*
$w_1 + w_2 + w_3 + w_4 + w_5 + w_1\_w_2\_w_3\_w_4\_w_5$
For example if the query is animalatzoo, the query used for Google image is animal + zoo + animal_zoo.

All results are reported in Table 1. As we can see, the approach based on visual features (Run 1) gives better results than those obtained when textual features are used (Run 2). This confirms that the assumption made about the visual representation of the query (using the RMAC descriptors of the top 3 images) is admissible. The comparison of the results of Run 3 and 4 shows that using the tags (with the proposed approach) was not able to bring significant improvements in comparison to the simple combination of visual features with username grouping. Finally, using images retrieved by Google engine (Run 5) outperforms significantly the results of Run 3 (visual + textual). This achievement leads us to reflect on the effectiveness of the proposed approach based on textual features. In order to achieve results close to those of Run 5, we should find a better solution for text analysis since there is no image query. Our future developments will mainly focus on exploiting different approaches to improve image retrieval based on metadata.

4 CONCLUSION

In this paper we presented a detailed description of the approach proposed to address the task of retrieving diverse social images. The proposed approach achieves promising results and shows the potential of automatic techniques in improving both precision and diversity. The comparison of the different runs shows that contrary to what we expected, textual information is outperformed by visual information. This observation raises some questions regarding the proposed approach and the quality of the provided metadata. We plan to investigate these questions in more detail and bring new solutions in our future work.

REFERENCES

[1] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[2] Bogdan Boteanu, Ionut Mironica, and Bogdan Ionescu. 2016. LAPI@ 2015 Retrieving Diverse Social Images Task: A Pseudo-Relevance Feedback Diversification Perspective.. In *MediaEval*.

[3] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. 2011. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*.

[4] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, and others. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Kdd*, Vol. 96. 226–231.

[5] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. 2016. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision* (2016), 1–18.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[7] Seddati Omar, Dupont Stéphane, Mahmoudi Saïd, and Pariyaan Mahnaaz. 2017. Towards Good Practices for Image Retrieval Based on CNN Features. In *International Conference on Computer Vision Workshop (ICCVW)*.

[8] Arora Sanjeev, Liang Yingyu, and Ma Tengyu. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *Proceedings of ICLR 2017*.

[9] Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation* 28, 1 (1972), 11–21.

[10] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. 2015. Particular object retrieval with integral max-pooling of CNN activations. *arXiv preprint arXiv:1511.05879* (2015).

[11] Sabrina Tollari. 2016. UPMC at MediaEval 2016 Retrieving Diverse Social Images Task. In *MediaEval 2016 Workshop*.

[12] Greg Corrado Jeffrey Dean Tomas Mikolov, Kai Chen. 20013. Efficient Estimation of Word Representations in Vector Space.. In *arXiv preprint arXiv:1301.3781*.

[13] Maia Zaharieva. 2016. An Adaptive Clustering Approach for the Diversification of Image Retrieval Results.. In *MediaEval*.

[14] Maia Zaharieva, Bogdan Ionescu, Alexandru Lucian Gînscă, Rodrygo L.T. Santos, and Henning Müller. 2017. Retrieving Diverse Social Images at MediaEval 2017: Challenges, Dataset and Evaluation. In *Proc. of the MediaEval 2017 Workshop, Dublin, Ireland, Sept. 13-15, 2017*.