User-Centred Image Retrieval Model

Allan Hanbury [*], Mihai Lupu[*], Pinar Duygulu[†], Adrian Popescu[‡], Adrian Iftene[§]

[*]Vienna University of Technology, Austria

[†]Bilkent University, Turkey

[‡]CEA, LIST, LVIC France

[§]UAIC, Romania

Contacts: lupu@ifs.tuwien.ac.at, duygulu@cs.bilkent.edu.tr, adrian.popescu@cea.fr, adiftene@info.uaic.ro

`MUCKE Project, Deliverable 5.2`

07/07/2014

Contents

Abstract

The user-centred image retrieval model plays an important role in demonstrating the viability of MUCKE framework through integrating the user credibility and concept similarity in a practical setting. The retrieved results are desired to be diverse as well as being representative. This is challenging because it involves the accurate balancing of similarity and dissimilarity, which we aim to achive through our methods developed within MUCKE. We introduce user credibility estimation in the retrieval process in order to favour the presentation of results uploaded by highly trusted users. This deliverable describes efforts in multiple directions to address these challenges.

1 INTRODUCTION

Although image retrieval has been a well studied area, with the popularity of social photo sharing websites it became more important.

In today's world image sharing applications are being used extremely. For example, users of Facebook upload 350 million photos[1] each day and it is said to be equal to the number of photos have been taken during 19th century in total[2]. Given that large number of images, search engines become more important than ever in order to produce good quality search results.

Since there is no quality control of the photos and of their annotations, retrieved images for a given query are usually not sufficiently relevant. One other major challenge is the diversification of the results while keeping the precision high. Most of the existing approaches focus on the relevance of the results without considering the very similar results as a drawback. However, especially since the users prefer to browse only a small set of the retrieved results, they expect to retrieve not only representative but also diverse results covering the query in the best way.

MUCKE introduces an image retrieval model which integrates user credibility estimation, multimedia concept similarity and multimedia fusion as central pieces of the framework in order to produce representative and diversified search results. The feasibility of the proposed methods are demonstrated by contributing to the MediaEval Retrieving Diverse Social Images Task in 2013 and 2014.

Introduced in 2013, the MediaEval Retrieving Diverse Social Images Task [8] was proposed to foster the development and evaluation of methods for retrieving diverse images of different point of interest. The problem of result diversification in social photo retrieval is addressed with a use case scenerio where a tourist tries to find more information about a place to visit. Provided with a ranked list of photos of a location retrieved from Flickr. It is required to exploit the provided visual and textual information to refine the noisy and redundant results by selecting only a sub-set of photos. In 2014, information about user annotation credibility is also provided.

As MUCKE, we participated in the MediaEval diversity task as a set of sub-teams each contributing with different methods. The following will summarise the proposed methods in detail.

2 CONTRIBUTIONS FOR MEDIAEVAL 2013 RETRIEVING DIVERSE SOCIAL IMAGES TASK

The Mediaeval 2013 Retrieving Diverse Social Image Task addresses the challenge of improving both relevance and diversity of social photos in a retrieval task. We propose a clustering based technique that exploits both textual and visual information. We introduce a k-NN inspired re-ranking algorithm that is applied before clustering to clean the dataset. After the clustering step, we exploit social

[1] http://www.businessinsider.com/facebook-350-million-photos-each-day-2013-9

[2] http://blog.1000memories.com/94-number-of-photos-ever-taken-digital-and-analog-in-shoebox

cues to rank clusters by social relevance. The following will describe the details.

2.1 DATASET AND FEATURES

2.1.1 DATASET

The dataset used in the task is collected from Flickr. Annotations are usually limited by a few number of tags and therefore incomplete [14]. Flickr images also come with social cues, such as user ID. We exploit both textual and visual features as well as social cues.

2.1.2 TEXTUAL FEATURES

We exploit a classical TF-IDF weighting scheme to model textual information associated to points of interests (POIs). Different re-weighting schemes were experimented and the best results were obtained when we took the square root of TF-IDF scores. The dimension of the model is equal to the number of unique tags associated with each POI: it is usually in the range of hundreds. L1-normalization is applied to textual features.

2.1.3 VISUAL FEATURES

We exploit visual features in order to overcome the sparsity of textual annotations. We use the Histogram of Gradients (HOG) in our experiments. In addition, we extracted GIST [22] and bags of visual words (BOVW) based on dense SIFTs [17] that proved to be efficient in large scale image retrieval. Dense SIFT descriptors are extracted using a codebook of size 1024. A spatial pyramid model [16] with two levels is used and the resulting feature size is 8192. HOG, GIST and BOVW features capture different low-level characteristics of images and they can be combined to have more comprehensive visual representations. When combined, all features were L1-normalized in order for each of feature to have the same contribution, regardless of their size.

2.2 RANKING

2.2.1 RESULT RERANKING

We use the features to rank the images based on their similarities. However, resulting ranked list is not satisfactory. We introduce a k-NN inspired approach that exploits visual and social cues to rerank the results.

We consider all the images of the POI as a positive set and constructed a negative set of the same size by sampling images of other POIs from the collection. Then we compared the HOG features of each image to all other images' features from positive and negative sets and retained the top 5 most similar results. We counted the number of different users that contributed to positive top five neighbors and, then the number of positive top five neighbors and the average distance to the first five positive neighbors. Images were then ranked by cascading the three scores described and we used 70%, 80% and 90% of the results as an input for the clustering process. The best results obtained on the devset and the best scores were obtained with 70% of the initial list retained and this threshold was retained for clustering as will be described in the following.

2.2.2 CLUSTERING

In clustering process k-means++ algorithm is used to cluster the images of a topic using previously mentioned feature types. Different numbers for k values are tested in experiments such as 10, 15 and 20. K value is selected as 15 since the official evaluation metrics are considered at 10 by the task organizers and selecting 15 as the k value provides us better results.

2.2.3 CLUSTER AND RESULT RANKING

Clusters are not all born equal and we need to be able to rank them by probability of relevance of contained images. Inspired by [15], we exploit social cues for cluster ranking and propose a simple scheme that is based on user and date information. For each cluster, we count the number of different users that contributed to it and the number of different dates when photos were taken. The first count aims to prioritize clusters that are socially diverse while the second count aims to surface clusters that are temporally stable. Then we calculate the product of these two counts and consider it as a social ranking score. To break ties, we also use the number of images present in each cluster.

For each POI, we retain only the top ten clusters obtained with the cluster ranking procedure and then diversify images by choosing one image from each cluster by descending similarity to the cluster centroid.

2.3 RESULTS AND DISCUSSION

To address the diversified social image retrieval problem, participants are asked to submit different types of runs. In this work four different runs are proposed. These runs produced by using different types of features and their combinations on the same dataset. The submitted four runs are described below:

- RUN1 is produced using only textual features.
- RUN2 is based on visual features. We concatenate HOG and GIST features.
- RUN3 is produced using a combination of textual features and GIST features. Visual and textual features are concatenated and to produce feature vectors. Linear weighting is used with 0.7 and 0.3 weights that are given to visual and textual features respectively. These weights were empirically chosen by testing different combinations on the devset.
- RUN4 is similar to RUN3, the only modification being the replacement of HOG features by BOVW features.

TABLE 1: GEOLOCATION PRECISION AT DIFFERENT SCALES

Run name	CR@10	P@10	F1@10
RUN1	0.3869	0.7333	0.489
RUN2	0.3892	0.7243	0.4905
RUN3	0.3848	0.7272	0.4868
RUN4	0.3742	0.7161	0.4753
Flickr baseline	0.3649	0.7883	0.4693

The results in Table 1 show that there are only small differences between the four runs. All submitted runs were slightly better than the Flickr baseline but the gain is not very significant. We were surprised to see how well the textual run (RUN1) performed compared to visual and multimodal runs since we had expected visual diversification to work better for POIs, which usually have a limited number of visual aspects. The performance drop from RUN 3 to RUN 4, due to the replacement of HOG features with BOVW features also came as a surprise since the latter usually work well for retrieval processes over visually diversified datasets.

3 CONTRIBUTIONS OF CEA LIST FOR FOR MEDIAEVAL 2014 RETRIEVING DIVERSE SOCIAL IMAGES TASK

CEA LIST's efforts are channeled towards exploiting visual information and the use of credibility in the diversication process. We first describe a couple of pre-filtering techniques followed by an image retrieval method that boosts precision. Next, we describe how to predict a user's credibility score and we propose a user based image filtering approach. After we show how we improve diversity by clustering and cluster ranking, we finally describe the submitted runs and discuss the results we obtained on the testset.

3.1 AIMING FOR PRECISION

3.1.1 INITIAL PRE-FILTERING

We use two filtering steps with the goal to eliminate noise form the image lists. Similar to [11], we eliminate geotagged images that have a distance from the POI higher than 1 km. The second filter is a restriction on the presence of faces in images. We use the standard OpenCV[3] algorithm to perform face detection and we eliminate images having a face coverage ratio higher than 0.4. The distance threshold and the one for the percentage of faces are determined on the devset. We keep the same pre-filtering steps for all the runs.

[3] http://opencv.org/

3.1.2 IMAGE RETRIEVAL

Following the latest advances in computer vision, we use Caffe [12], a powerful CNN-based feature, to extract representations for the images in the collection, as well as the Wikipedia image examples. Following a standard content based image retrieval approach, we rank the images for each topic by the average cosine similarity between the retrieved image and all of the example images.

On the devset, we obtain a P@20 of 0.966 when doing retrieval with the Caffe features. This represents a signi can't improvement over the Flickr ranking (P@20 = 0.831) and LBP3x3 (P@20 = 0.816), the descriptor provided by the organizers which gives the best performances in visual retrieval.

One drawback of this method is the strong trade of between precision and cluster recall. Although P@20 on the devset is high, we get a CR@20 of 0.293, leading to a F1@20 of 0.438. This problem is directly approached by first selecting images found in different clusters, as described in the following.

3.2 LISTENING TIO SOCIAL CUES

3.2.1 PREDICTING USER CREDIBILITY

We exploit the credibility set to train a regression model that predicts a user's credibility score from the provided features. We perform model selection and parameter tunning by 5-fold cross-validation (cv) on the credibility set and we evaluate the performance of the predictions by Spearman's rank correlation coefficient with the ground truth credibility values. The highest cv correlation (0.47) is obtained using gradient boosting regression trees with a Huber loss and 100 estimators. By comparison, the highest correlation of an individual feature (visual score) is 0.36. The gain in regards to the Spearman score is als reflected on the competition metrics. When fixing the rest of the parameters and using the predicted credibility scores instead of the provided visual credibility feature, F1@20 increases from 0.61 to 0.632 on the devset.

3.2.2 USER SELECTION

For each topic, we first keep a subset of users that have contributions in the top n images found in the ranking produces by the image retrieval process described above. Then, as an extra filter, in our final ranking we retain only images coming from the selected user set. Given the good precision of image retrieval, we have a high confidence that images found in the top of the ranking are relevant. This gives us an ad-hoc topical expertise insight about the users responsible for those images. We tune n on the devset and fix it at 20. For comparison, when not using a user based filter, the F1@20 score drops from 0.632 to 0.597. We also tried a similar approach by retaining contributions from top users ranked according to the credibility score but this did not improve the results. This result hints at the need for a topic specific credibility score.

3.3 IMPROVING DIVERSITY

Building on previous works, we combine a more traditional clustering approach for diversification with the use of social cues [5].

3.3.1 CLUSTERING

We first perform k-Means clustering on the complete set of images. To ensure a stable cluster distribution, we initialize the centroids by uniformly selecting images from the ranking produced after image retrieval. For example, the i-th cluster will have as initial centroid the image found on the position $(i-1)xn/k$, where k is the desired number of clusters and n is the number of images in the ranking. After validation on the devset, k is set to 30.

3.3.2 CLUSTER RANKING

We leverage the social component of this task by ordering the clusters based on the average credibility score of the users that contribute with images in the cluster. For the runs that do not permit the use of credibility, we rank the clusters according to the number of unique users represented in each cluster. In the case of a tie, we prefer the cluster that has the best ranked image after visual retrieval. Our final ranked list is obtained by selecting from each cluster at a time the image that is best placed in the visual retrieval ranking.

3.4 RESULTS AND DISCUSSIONS

We submitted five different runs at this year's Retrieving Diverse Social ImaImages Task [10]. Our submissions are briefly described below:

RUN1 uses the provided LBP3x3 visual descriptor for image retrieval and clustering. The clusters are then ranked based on the number of users represented in each cluster.

RUN2 is a purely textual one. We concatenated the title, tags and description of the photos to calculate the text similarity. As text pre-processing phase, we de- compounded the terms by applying a greedy approach using the dictionary which is created by all the words in the text. In the next step, in order to disambiguate the places, we expand the queries using the first sentence of Wikipedia. After testing several language models, using a semantic similarity approach based on Word2Vec [21] gave the best result. We trained a model on Wikipedia and then used the vector representation of words to calculate the text similarity of the query to each photo. In additional to the text similarity, we extracted three binary attributes: (1) if the photo had any views, (2) if the distance between a photo and the POI is greater than 8 kilometers, and (3) if the description length has more than 2000 characters. All features were then used in a Linear Regression model in order to re-rank the list. Finally, following [25], in order to diversify the ranking, we iterate over the initial re-ranked list and keep one image from each user at each iteration.

TABLE 2: RUN PERFORMANCES WITH THREE OFFCIAL METRICS.

Run name	F1@20	P@20	CR@20
RUN1	0.5182	0.7313	0.4103
RUN2	0.5346	0.8089	0.4084
RUN3	0.5525	0.798	0.4335
RUN4	0.5243	0.7378	0.4157
RUN5	0.571	0.7931	0.4563

RUN3 is a fusion between RUN1 and RUN2. Since the scores for visual and textual rankings are not in the same range, fusion is performed based on the ranks of the images in the two initial rankings. More specifically, we perform a linear weighting in which the individual ranks are given a weight of 0.5. Other weighting have been tested but the results remain quite stable in the range 0.3 - 0.7, a result which accounts for the robustness of the proposed fusion.

RUN4 is similar to RUN1 with the single difference lying in the use of credibility for cluster ranking.

RUN5 is obtained using the Caffe visual descriptor for image retrieval and clustering and predicted credibility scores for cluster ranking.

Our textual run (RUN2) is the single one in which we do not use clustering to improve diversity. This reflects across metrics, as it can be seen in Table 2. Although it performs well in terms of F1@20, this run is placed at oposite poles when looking at the other metrics. It has the highest P@20 and the lowest CR@20. The usefulness of credibility can be best observed when comparing RUN1 and RUN4. They share the same confuguration with the sole exception being the use of the predicted credibility scores for cluster ranking in RUN4. Although the difference is not as significant as on the devset, we can see a slight improvement of F1@20.

4 CONTRIBUTIONS OF TUW FOR FOR MEDIAEVAL 2014 RETRIEVING DIVERSE SOCIAL IMAGES TASK

This section describes the efforts of Vienna University of Technology (TUW) in the MediaEval 2014 Retrieving Diverse Social Images challenge. Our approach consisted of 3 steps: (1) a pre-filtering based on Machine Learning, (2) a re-ranking based on Word2Vec, and (3) a clustering part based on a ensemble of clusters. Our best run reached a F@20 of 0.564. We employed a distinct set of methods for each run. In the following, we explain all the approaches and on Table **??** we show what was used in each run.

4.1 PRE-FILTERING

We employed a pre-filtering step to filter out potential irrelevant pictures. For the 2014 development set, we calculated that 70% (6256) are relevant images, while 30% (2667) were not. The goal of this step is to increase the percentage of relevant images. After applying a Logistic Regression classifier trained on the 2013 data, we could reach a ratio of 74% (5780)/26% (2026), removing a total of 1117 images. As features, we used the distance between images and POIs, number of views, length of descriptions and titles, images' license, part of the day (morning, afternoon, night) and the number of times the POI appeared in the title and descriptions of an image.

4.2 RE-RANKING

For re-ordering the results, we used the title, tag and description of the photos. For text pre-processing, we de-compounded the terms using a greedy dictionary based approach. In the next step, we expand the query using the first sentence of Wikipedia which helps for place disambiguation. We tested four document similarity methods based on Solr[4], Random Indexing[5], Galago[6] and Word2Vec [**?**]. Among all, we found the best result using a semantic similarity approach based on Word2Vec.

Word2Vec provides vector representation of words by using deep learning. We used the Word2Vec library[7] and trained a model on Wikipedia and then used the vector representation of words to calculate the text similarity of the query to each photo.

Similar to the pre-filtering, we extract three binary attributes: Number of views, distance between photos and POIs if it is more than 8 and description length if it is more than 2000 characters. All features were applied in a linear regression model in order to re-order the list.

4.3 CLUSTERING

We worked on three methods for clustering, all based on similarity measures. They share the idea of creating a similarity graph (potentially complete) in which each vertex represents an image for one point of interest, and each edge represents the similarity between two images. Different similarity metrics and different set of features can be used. Next, we explain each algorithm and how we combined them.

Metis:

The first approach, called Metis [13], tries to collapse similar and neighbor vertices, reducing the initial graph to a smaller one (known as coarsening step). Then, it divides the coarsest graph into a pre-defined number of graphs, generating the clusters.

Spectral:

[4] http://lucene.apache.org/solr/

[5] https://code.google.com/p/semanticvectors/

[6] http://sourceforge.net/p/lemur/galago

[7] https://code.google.com/p/word2vec/

Spectral clustering [27] can also be seen as a graph partitioning method, which measures both the total dissimilarity between groups as well as the total similarity within a group. We used the Scikit-learn[8] implementation of this method.

Hierarchical:

Hierarchical clustering [29] is based on the idea of a hierarchy of clusters. A tree is built in a way that the root gathers all the samples and the leaves are clusters with only one sample. This tree can be built bottom-up or top down. We used the bottom-up implementation from Scikit-learn[9].

Merging:

We found that the clustering methods were unstable as modifications in the filtering step caused a great impact in the clustering step. Therefore, we decided to implement a merging heuristic, which takes into account different point of views from each clustering method and/or feature set, being potentially more robust than using one single algorithm.

First, we run each clustering algorithm using a different feature sets (for example, HOG, CN, and text similarity) and different distance measures (in all experiments we used both cosine and Chebyshev) for each POI. It generates a great number of possible cluster results (3 algorithms × 3 feature sets × 2 measures = 18 possible ways to make clusters). We then created a re-ranking heuristic based only on the frequency that two documents occur in the same cluster.

The goal of the re-ranking based on the cluster results is to move all documents from the original list (Flickr ranking) to a re-ranked list. We start the procedure moving one pivot document (in this work, the top ranked document in the original list) to the re-ranked list. Then, for each document D_i from the original list, we count the number of times that D_i occurred together with each element in the re-ranked list. If any of these frequency values is bigger than a pre-defined $Max\_Threshold$ (6 out of 18, for example), we do not move D_i to the re-ranked list, because D_i was co-occurred frequently with another document that is already in the re-ranked list. However, if D_i was not frequently seen with any other document, than we move D_i from the original list to the end of the re-ranked list. After trying to move all the documents from the original list to the re-ranked list, we increase the $Max\_Threshold$, so we can accept an image even that a greater number of clustering methods assign that image to the same cluster of another image. Usually this is the case of the least ranked element. The algorithm stops when all documents (or the 50 first) are moved from the original list to the re-ranked one. We also used a $Min\_Threshold$ and a $Mean\_Threshold$, but other measures, such as the mean or any percentile could be easily employed as well.

4.4 CREDIBILITY

Our approaches were based on Machine Learning: we trained a Logistic Regression classifier to learn if a document is relevant or not based on the credibility data (used only face proportion, location

[8] http://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html

[9] http://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

similarity, upload frequency and bulk proportion). We tested two methods: we (1) filtered out documents set as irrelevant for Run4 and (2) moved to the bottom of the list irrelevant documents for Run5.

4.5 EXPERIMENTS

We submitted all 5 runs, varying on the use of pre-filtering, the re-ranking method, the clustering approach and the use of credibility. Details are shown on Table **??** and the results are shown on Table **??**.

4.6 CONCLUSION

Our experiments show that an ensemble of clusters can be a robust way to diversify results. Unfortunately our re-rank method did not work in the test set as well as it did in the development set. Last, the use of credibility also seem to have overfitted the development data, not being effective for the development set.

5 CONTRIBUTIONS OF BILKENT FOR FOR MEDIAEVAL 2014 RETRIEVING DIVERSE SOCIAL IMAGES TASK

This section describes the approach proposed by Bilkent - RETINA team for the Retrieving Diverse Social Images task of MediaEval 2014 [9]. We develop a framework which first removes outliers using one-class support vector machines (SVM) to improve relevance. Second it clusters the eliminated set and retrieves the centroids to diversify the results. We tried to exploit visual only features during our experiments. For the first run we used the provided visual features and for the second run we used well known visual features like SIFT [18] and GIST [23]. The following will explain the details.

5.1 PROPOSED APPROACH

Our method can be summarized in 4 steps as shown in Figure 1, namely:

Step 1: *Feature extraction*

In this step we compute visual features for each image of each location. Some of the features are provided by the task and 2 of them are extracted by our team.

Step 2: *Outlier removal*

In order to increase number of relevant images for each location in the dataset, we apply an outlier removal procedure. This procedure promisingly chop off some of the irrelevant images from the dataset and increase the *P* and *F1* scores.

Step 3: *Clustering*

After the outlier removal step, in order to increase the diversity score we apply k-means clustering to the remaining images at each location.

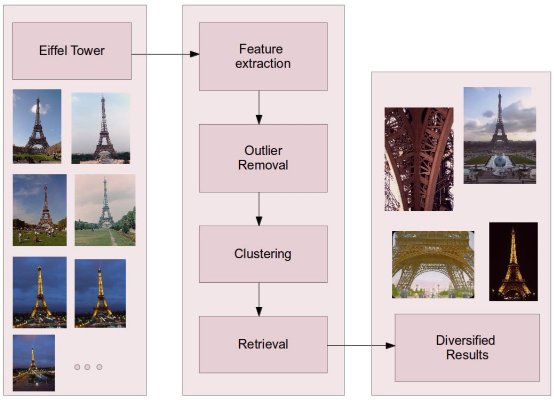

FIGURE 1: OVERALL FRAMEWORK STRUCTURE. WHEN THE IMAGES RELATED TO A SPECIFIC LOCATION ARE GIVEN AS INPUT, OUR FRAMEWORK PRODUCES DIVERSIFIED RESULTS FOR THAT LOCATION.

Step 4: *Retrieval*

In the retrieval step we select cluster centroids that we obtain in the previous step. Each centroid should represent a different aspect of a given location so that it is aimed to get a good diversification results.

5.2 VISUAL FEATURES

The task organizers provide us with 6 visual descriptors (CM, CN, CSD, GLRLM, HOG, LBP) out of which 4 have also a spatial pyramid representation (CM, CN, GLRLM and LBP). We sought for the best combination of these features using provided devset images. We found out that best results are obtained when all these features are combined. So we concatenate all these 10 visual descriptors and come up with a feature vector of 945 dimensions for each image (i.e., descvis). Then we normalize each feature vector to zero mean and unit variance.

We also extracted other visual features like GIST and bag of visual words (BOVW) representations using dense SIFT features [18, 23]. We use these extra features while constructing the fifth run of the challenge. GIST features are 512 dimensional global features and they are useful in capturing the scene information in images. It is important to capture and differentiate scenery information in order to boost diversity of the results.

In order to compute dense-SIFT descriptors we use *vlfeat*'s standart feature extactor tool [28]. First we resize each image to a fixed size of 200 by 200 pixels and then we obtain 128 by 5776 dimensional SIFT features per image. In order to create a pool of descriptors we randomly sample 100 descriptors from each image and then we apply k-means algorithm with 'plusplus' option. We try 3 different k values (e.g., 600, 800 and 1000). According to the performance on devset, we choose k of k-means as 1000 and it becomes the volume of our visual words dictionary. Using this dictionary, we quantize each image to 1000 dimensional feature vectors.

5.3 OUTLIER REMOVAL

We use SVM to find out the outliers and construct a subset of images per location which are more relevant than the initial set. Our method is similar to [19] but we use a fixed set of negative examples for each of devset and testset which are selected in the following ways. For devset images we picked 2 random images from each of the 30 locations, for testset images we select 60 random images from each of the 123 locations considering at most 1 image from each testset location. Then for each location, similar to cross validation, we select 60 random positive images and first train and then classify using one-class SVM, and repeat this procedure 10 times consecutively. Finally we select the model which scored the highest accuracy assuming that this model provides the best seperation. We use this process for each location, using the same negative examples at each step but with different positive examples. We use a *quadratic* kernel while experimenting with SVM because our features are dense vectors so that they are not easily seperable by linear kernel functions. We observed on the devset that as the result of outlier removal process, we get rid of some of the irrelevant images and obtain a higher relevancy score for each location.

5.4 CLUSTERING AND RETRIEVAL

After outliers are removed we cluster the images of each location using a k-means algorithm. On the devset we try 2 different K values. First we select K as 25, because we observed that each location has at most 25 subclasses in their diversity subgroups. Second we select K as 50, because that was the maximum number of images required to be retrieved. The latter method, over clustering, seemed to work better in devset so that we report our test set results using K as 50.

After we compute cluster centroids, we simply retrieve images which are closest to the centroids. We apply *k nearest neighbor* method with Euclidean distance and search for the nearest neighbor for each centroid. While computing nearest neighbor we pay great attention to retrieve unique neighbors for each cluster centroid.

Results from devset are shown in Table 3. One may observe that SIFT-BOVW [18] features works better than default features. The reason is that local descriptors are generally works better to capture similarities among images so that each cluster becomes more coherent. GIST [23] features also perform better than the default features and perform similar to SIFT-BOVW features. Results from our 2 submissions, namely *Run#1* and *Run#5*, can be found in Table 4. Similar to devset results, using SIFT-BOVW we obtain better results from *Run#5* than *Run#1*.

5.5 DISCUSSIONS

We showed that it is possible to obtain competitive results using only visual features. Our framework first eliminates the outliers and then using clustering it tries to leverage the diversity to the retrieval results. However it is obvious that one can improve the scores by utilizing more information into our framework like textual features, credibility scores.

TABLE 3: RESULTS ON DEVSET USING PROVIDED FEATURES, GIST AND SIFT-BOVW.

Feat. name	P@20	CR@20	F1@20
descvis	0.7139	0.3813	0.4863
GIST	0.7209	0.3798	0.5037
SIFT-BOVW	0.7167	0.3933	0.5013

TABLE 4: OFFICIAL RESULTS ON TESTSET.

Run#	P@20	CR@20	F1@20
1	0.6809	0.375	0.4758
5	0.7228	0.387	0.4966

6 UAIC - IMAGE AND USER PROFILE-BASED RECOMMENDATION SYSTEM

A great variety of websites try to help users in finding items of interests by offering a list of recommendations. It has become a function of great im-portance, especially for online stores. This paper presents a recommendation sys-tem for images which works with ratings to compute similarities, and with social profiling to introduce diversity in the list of suggestions. The image recommendation system presented in this part uses similarity between items, similarity between users and social profiling to predict what other images a user might enjoy.

6.1 RECOMMENDATION SYSTEMS

Recommendation systems represent a class of Web applications that predict user responses to options [26]. They automatically predict the information or items that may be of interest to a user and help in overcoming information overload by personalizing suggestions based on likes and dislikes. Such systems can be found in many online sites, especially online stores (e.g. Amazon, eBay, Barnes & Noble, IMDb, YouTube), making it much easier to explore the various available options and to find items of interest. They are valuable as they reduce the cognitive load on users, help with the Big Data problem and play a part in introducing quality control.

In a recommendation system there are two types of entities: *users* and *items*. Users have preferences for certain items and it is these preferences that such a system must identify. The data available to the system is represented as a utility matrix [26]. It contains, for each user-item pair, a value that represents the degree of preference of the user for the respective item. This matrix is generally sparse and the goal of a recommendation system is to predict the values in the blank entries. However, it is not always necessary to predict every such blank, but only those entries in each row that are likely to contain high values [26].

There are a number of different technologies used by recommendation systems, but two broad groups [26] can be distinguished. *Content-based* systems examine the features or properties of the suggested items. They recommend items that are similar in content to other items the user has previously expressed interest in. *Collaborative filtering* systems provide recommendations using various similarity measures between users and/or items. They collect human judgements [7] in the form of ratings for items and exploit the similarities and differences of user profiles when selecting what to suggest. The recommended items for a user are the ones preferred by other similar users.

In constructing our system, we approached the problem in a manner similar to [20], by combining item-based and user-based collaborative filtering and allowing each of these techniques to compensate when the other produces few or no results. We have also created recommendations based on social data, in order to encourage diversity and avoid the echo chamber effect.

6.2 UAIC SYSTEM

6.2.1 GATHERING DATA AND CREATING IMAGE PROFILES

The data on which our system operates was gathered by asking 78 students to tag and rate on a scale of 1 to 5 (5 - I like it very much; 4 - I like it; 3 - Neutral; 2 - I dislike it; 1 - I dislike it very much) a set of 100 images. The tags were lemmatized using Stanford CoreNLP, a suite of natural language analysis tools created by The Stanford Natural Language Processing Group [3]. The stopwords were eliminated and, for each image, the list of most frequent tags (i.e. with frequency higher than 5), in decreasing order, constituted its profile. There were, on average, 13.64 tags per image.

6.2.2 RECOMMENDATIONS BASED ON IMAGE SIMILARITY

We created an undirected graph, as depicted in Figure **??**, where the vertices are either images or tags. There are edges from images to tags and between tags. There are no edges between images. The edges represent different types of relationships between vertices and have weights between 0 and 1. An image may be connected to multiple tags and a tag may be connected to multiple images. We call this type of relationship an *Annotation* and its weight depends on the position of the tag in the list for the image and the total number of tags associated with that image.

There are nine types of relationships between two tags:

- *Subword* (weight 0.8) - A tag is a word contained inside another tag that is a phrase;
- *Common words* (weight 0.4)- The two tags are phrases that contain common words;
- *Attribute* (weight 0.7), *Nominalization* (weight 0.7), *Hypernym* (weight 0.6), *Similar to* (weight 0.3), *See also* (weight 0.2) - They were extracted using WordNet 3.0 [4] and have the same meaning as the respective pointer types [1] in WordNet. If tag t_1 and tag t_2 are in one of these relationships, it means that one of the synsets of t_1 has a pointer of this type to one of the synsets of t_2, or the other way around

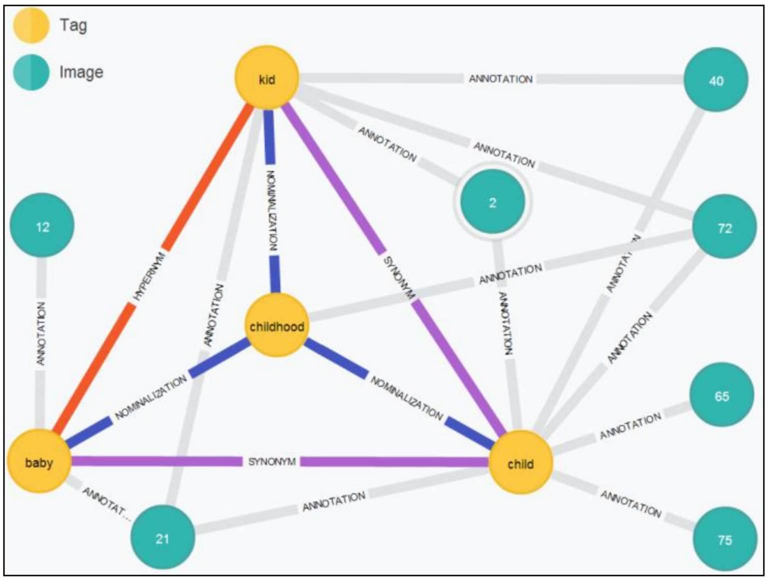

FIGURE 2: A SMALL PORTION OF THE IMAGES AND TAGS GRAPH, WHERE THE NODES CONTAINING NUMBERS ARE IMAGES (THE NUMBERS BEING THEIR ID) AND THE NODES CONTAINING TEXT ARE TAGS.

- *Synonym* (weight 0.9) - If tag t_1 and tag t_2 are in this relationship, it means that t_2 belongs to one of the synsets of t_1, or the other way around;
- *Indirect synonym* (weight 0.15) - If tag t_1 and tag t_2 are in this relationship, it means that their synsets have words in common.

Using this graph, which was stored using Neo4j, a highly scalable and robust native graph database [2], it is possible to start in an image node and reach other image nodes by passing through several tag nodes. Along a path (e.g. 2 → child → baby → 12), the weights of the edges are multiplied and the value with which an image node is reached represents the similarity (on this path) between the image from which we started and the image to which we arrived. For two images, their final similarity score is the highest score of all possible paths between them.

For a user u the recommendation process works as follows:

- We take all images that u rated with 4 ("I like it") or 5 ("I like it very much");
- For each such image, we look at the list of similar images, discard those that u has already rated and multiply the rating with the similarity score, thus obtaining a prediction for how the user would rate the new image;
- If there are multiple predictions for an image, we retain the maximum value;
- The images that are viable for recommendation are those whose predicted rating is higher than 3.5.

6.2.3 RECOMMENDATIONS BASED ON USER SIMILARITY

A popular method for doing user-based collaborative filtering is to regard the problem as a machine learning one and use a classifier such as k-NN. This means that the recommendations for a user will be an aggregation of what the k users most similar to them have liked. When computing the similarity between two users, we look at the images that they both have rated and compare the two vectors of ratings. For this purpose, we have tried two metrics: Pearson's correlation coefficient (see [5], chapter 7) and cosine similarity (see [26], section 3.5.4).

The list of recommendations for user u is created as follows:

- The list of ratings given by u is extracted;
- The list of recommendation scores is obtained. These are recommendations coming from the k users most similar to u, for images not rated by u. Let N be the set of k-nearest neighbors for u and N_i the subset of these neighbors that have given ratings to an image i. The recommendation score for i is computed as shown in (1).

$$rec\_score(i,u) = \frac{\sum_{n \in N_i} rating(n,i) \cdot similarity(u,n)}{\sum_{n \in N} similarity(u,n)} \tag{1}$$

- From this list, based on thresholds for minimum score and maximum number of results, the final list of recommendations is created.

The process of creating recommendations depends on three parameters: rating sub-set size, minimum recommendation score and maximum number of recommendations. Experimenting with different values for these parameters, we did several simulations in order to establish the best thresholds and to decide on the metric for user similarity. In assessing the results, we used four criteria:

- *Accuracy* - What percentage of the recommendations are images the user likes. If R is the set of recommended images, the accuracy is given by (2);

$$accuracy(u,R) = \frac{|\{i \in R | rating(u,i) \geq 4\}|}{|R|} \tag{2}$$

- *Fault* - How bad the recommendations not liked by the user are. If R is the set of recommended images and $R_b \subseteq R$ is the subset of bad recommendations, the fault is given by (3). We divide by 5 (i.e. the highest rating) to obtain a value between 0 and 1.

$$fault(u,R) = (|R_b|/|R|) \cdot (1 - \sum_{i \in R_b} \frac{rating(u,i)}{5 \cdot |R_b|}) \tag{3}$$

- *Real rating mean* - The mean value of the ratings the user had given to the recommended images (which we ignored and tried to predict);
- *No recommendations given* - For some thresholds it is possible that no recommendations are given due to very small recommendation scores. This criteria is represented by the number of users (out of 78) for whom no recommendations could be made.

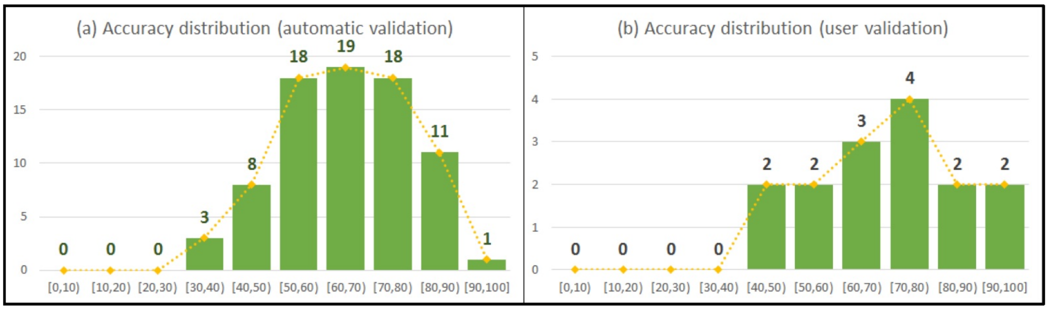

FIGURE 3: ACCURACY DISTRIBUTION FOR: (A) AUTOMATIC VALIDATION; (B) USER VALIDATION.

Based on the results, we decided to use Pearson correlation with a minimum score threshold of 4 (because it gives good results for all four criteria, while 4.25 and 4.5 might be quite restrictive and 4.5 introduces more cases of not being able to give recommendations). However, for the few exceptional occasions when it cannot be applied (e.g. the ratings given by the user have no variance), we use cosine similarity with a minimum score threshold of 3.875. For both cases, we take at most 10 results.

6.2.4 RECOMMENDATIONS BASED ON SOCIAL SIMILARITY

The approaches previously described will yield recommendations based on what the user has already liked. This could lead to an echo-chamber of what the user has explicitly expressed interest in. In an attempt to avoid this, we decided to include in our recommendation list some images that are not necessarily related to those that the user has previously given a high rating to, but have the potential to be of interest to them.

Thus, from the set of 78 users that we have used for training, 52 had Facebook accounts that we were able to access. We extracted their lists of interests and their rela-tionships on this social network. Recommendations were created by taking into account how similar their interests were (i.e. what ratio of user A's interests are also among user B's interests) and the distance between them in the network.

6.3 SYSTEM VALIDATION

The system was verified automatically, using the set of 78 users that have rated all 100 images. We have also tested the system on a set of 15 test users, different from the ones in the training set. As criteria for evaluation, we used accuracy, fault and real rating mean, as described in previous section. The results for accuracy are shown in Figure 3.

Automatic validation: Using the training set of 78 users, we have performed a cross validation of our system by taking each user and, in turn, making a random selection of their ratings, then comparing the recommendations of the system against the ratings that we previously discarded. The size of the selection was also random, between 8 and 30. We did 10 runs and, for each user, retained the average over all runs for each of the three evaluation criteria.

We believe the automated validation showed that our system works well. For almost 86% of cases at least half of the recommendations were good, the bad recommendations showed small faults, most of them between 0.1 and 0.25, and none of them greater than 0.35, while the real rating mean was in 77 out of 78 (98,72%) cases greater than 3 and in 60 out of 78 (76,92%) cases greater than 3.5, the average being 3.709.

User-based validation: The output of the system was also tested by 15 users differ-ent from the ones in our training set. They were each given a list of 10 recommendationsand offered feedback in the form of ratings on the same scale and with the same mean-ing as the one used when gathering the training data. Although there is little data com-pared to the automatic validation, the results are alike and prove that the system behaves in a similar manner.

For 86.67% of cases, the accuracy is at least 50%. The values for fault range up to 0.5, but most of them are between 0.1 and 0.3. The real rating mean was in 14 out of 15 (93.33%) cases greater than 3 and in 12 out of 15 (80%) cases greater than 3.5, the average being 3.78.

6.4 CONCLUSIONS

We have created an image recommendation system which uses similarity scores for items and users, combined with social profiling for diversity. It provides at least 50% good recommendations in about 86% of cases, very few lists of recommendations have a rating mean of less than 3, and about 76-80% have a rating mean of over 3.5.

The system was built using a small quantity of data, focusing on experimentation and the choices to be made between several possible approaches. It would be very in-teresting to examine the possibility of adapting it to work on a larger scale. One possible approach is to use a distributed system, running in the cloud [24], and using Map Reduce to compute results [6]. We are currently working on gathering more information, by using a collection of at least1000 images and asking for ratings from a group of about 500 students.

7 ADDITIONAL AUTHORS

- Navid Rekabsaz, Vienna University of Technology
- João R. M. Palotti, Vienna University of Technology
- Alexandru Lucian Ginsca, CEA LIST
- Anil Armagan, Bilkent University
- Ilker Sarac, Bilkent University
- Cristina Serban, UAIC

- Lenuta Alboaie, UAIC

References

[1] A glossary of wordnet terms, http://wordnet.princeton.edu/wordnet/man/wngloss.7wn.html.

[2] The neo4j manual v2.1.2, http://docs.neo4j.org/chunked/milestone.

[3] Stanford corenlp, http://nlp.stanford.edu/software/corenlp.shtml.

[4] Wordnet 3.0 reference manual, http://wordnet.princeton.edu/wordnet/documentation.

[5] S. Boslaugh. *Statistics in a Nutshell*. O'Reilly Media, Inc., 2012.

[6] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. In *OSDI'04: Sixth Symposium on Operating System Design and Implementation*, San Fran-cisco, CA, December 2004.

[7] J. Herlocker, J. Konstan, A. Borchers, and J. Reidl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, August 1999.

[8] B. Ionescu, M. MenÃ©ndez, and A. MÃ¼ller, H. andPopescu. Retrieving diverse social images at mediaeval 2013: Objectives, dataset and evaluation. In *Proceedings of MediaEval Benchmarking Initiative for Multimedia Evaluation*, 2013.

[9] B. Ionescu, A. Popescu, M. Lupu, A. L. Gînscă, and H. Müller. Retrieving diverse social images at mediaeval 2014: Challenge, dataset and evaluation. In *MediaEval 2014 Workshop, October 16-17, Barcelona, Spain, 2014.*

[10] B. Ionescu, A. Popescu, M. Lupu, A. L. Gînscă, and H. Müller. Retrieving Diverse Social Images at MediaEval 2014: Challenge, Dataset and Evaluation. In *Proceedings of the MediaEval 2014 Multimedia Benchmark Workshop*, October 2014.

[11] N. Jain. Experiments in diversifying flickr result sets. In *MediaEval 2013 Workshop*, 2013.

[12] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. `http://caffe.berkeleyvision.org`.

[13] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, Dec. 1998.

[14] L. S. Kennedy, S.-F. Chang, and I. V. Kozintsev. To search or to label?: predicting the performance of search-based automatic image classifiers. In *8th ACM international workshop on Multimedia information retrieval, MIR '06*, 2006.

[15] L. S. Kennedy and M. Naaman. Generating diverse and representative image search results for landmarks. In *17th international conference on World Wide Web, WWW '08*, 2008.

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[17] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 2011.

[18] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[19] H. Lukashevich, S. Nowak, and P. Dunker. Using one-class svm outliers detection for verification of collaboratively tagged image training sets. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pages 682–685. IEEE, 2009.

[20] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002)*, pages 187–192. Edmonton, Canada, July 2002.

[21] T. Mikolov. Efficient estimation of word representations in vector space. In *CoRR*, 2013.

[22] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 2001.

[23] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006.

[24] G. Pallis. Cloud computing - the new frontier of internet computing. In *IEEE Internet Computing*, volume 14, Issue 5, pages 70–73, 2010.

[25] A. Popescu. Cea list's participation at the mediaeval 2013 retrieving diverse social images task. In *MediaEval 2013 Workshop*, 2013.

[26] A. Rajaraman, J. Leskovec, and J. D. Ullman. *Mining of Massive Datasets*. Cambridge University Press, New York, NY, USA, 2011.

[27] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, Aug. 2000.

[28] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms.

[29] J. H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.