CERTH @ MediaEval 2014 Social Event Detection Task

Marina Riga, Georgios Petkos, Symeon Papadopoulos, Emmanouil Schinas, Yiannis Kompatsiaris
Information Technologies Institute / CERTH
6^{th} Km. Charilaou-Thermis
Thessaloniki, Greece
{mriga,gpetkos,papadop,manosetro,ikom}@iti.gr

ABSTRACT

This paper describes the participation of CERTH in the Social Event Detection Task of MediaEval 2014. For Challenge 1, we use a "same event model" to construct a graph on which we perform community detection to obtain the final clustering. Importantly, we tune the model to have a higher true positive rate than true negative rate, leading to significantly improved performance. The F1 score and NMI for our best run are 0.9161 and 0.9818, respectively. For Challenge 2, we developed probabilistic language models to classify events according to the criteria of the different queries. Our best run on Challenge 2 achieved an average F-score of 0.4604.

1. INTRODUCTION

The paper presents the approaches developed by CERTH for the two Challenges of the MediaEval 2014 Social Event Detection (SED) task. Challenge 1 asks for a full clustering of a collection of Flickr images, so that each cluster corresponds to a social event. Challenge 2 examines a retrieval scenario in which, given a set of social events, the goal is to determine those events that match particular criteria. More details about the task can be found in [3].

2. PROPOSED APPROACH

2.1 Overview of method in Challenge 1

Our approach for Challenge 1 utilizes what is termed the Same Event Model (SEM)[2]. The SEM takes as input the set of per modality similarities between two items and predicts how likely it is that these two items belong to the same event or not. Subsequently, a graph is constructed, in which the nodes represent the images to be clustered and the existence of an edge between a pair of nodes denotes the positive prediction of the SEM for the two respective images. Finally, a community detection algorithm is performed on the graph to obtain a full clustering. Moreover, in order to limit the number of evaluations of the SEM and make the approach scalable, we deploy a candidate neighbour selection step: for each image we utilize appropriate indices in order to obtain the most similar images according to each modality and evaluate the SEM only for them. This is a technique that is commonly referred to as blocking. This overall approach is similar to that of [5] and that which we deployed in last year's task [6]. Importantly though, we introduce a tweak which improves performance significantly. The key idea is that false positive and false negative predictions of the SEM are not equally important. More specifically, the average size of an event in the training set is roughly 20 images. In practice though, the set of candidate neighbours needs to be quite larger than the average. For instance, in our experiments we experimented with at most 500 candidate neighbours. The primary reasons for this is that a) the distribution of the sizes of the events is much wider and b) in large datasets one needs to consider a larger number of candidate neighbours in order to have higher confidence that the actual neighbours of some image appear in the set of candidate neighbours. Therefore, since the number of candidate neighbours will be much larger than the number of actual neighbours, and assuming that the classifier has been trained to achieve similar true positive and true negative rates, we can expect that the SEM will give a significantly larger number of false positive predictions than false negative predictions. Too many false positive predictions are likely to result in a lot of merged clusters as they will create too many incorrect edges in the graph. If on the other hand we opt for a higher true positive rate at the cost of a lower true negative rate (by increasing the classification threshold), we will have far fewer incorrectly merged clusters, but will also have some fragmented clusters. The way to deal with this problem is to increase the set of candidate neighbours. In our experiments, we observed that when increasing the threshold so that the true positive rate is 0.9999, the true negative rate does not drop below 0.95, which in practice appears sufficient for our purpose.

2.2 Overview of method in Challenge 2

In Challenge 2, we utilize regularized unigram language models [1] to classify clusters (or images in Run 5, as will be explained later) according to the given retrieval criteria (location, type of event, entities involved). For learning the language models for the event types and entities of interest we collected sets of images from Flickr using the relevant keywords that appear in the queries. Moreover, we retrieved an additional random collection of images, in order to learn a general language model that does not focus on any particular event type or entity, against which the type- or entity-specific language models are compared. For some cluster (or image) i the comparison is performed by computing the ratio of the probability given by the specific language model $p_{specific}(i)$ over the probability given by the general language model $p_{general}(i)$; if the ratio is above some threshold θ, then we assign the event (or image) as matching the examined criterion. In a second variation we utilize a language model that has trained both with the type and entity specific datasets and the general dataset and com-

MediaEval 2014 Workshop, October 16-17, 2014, Barcelona, Spain

	Challenge 1			Challenge 2 Average scores			Challenge 2, F1 per query									
Run	**F1**	**NMI**	**Div.**	**Recall**	**Precision**	**F1**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0.4514	0.7594	0.4498	0.6101	0.3458	0.3431	0.6207	0.6588	0.2137	0.2694	0.8193	0.1524	0.4578	0.0868	0.1375	0.0145
2	0.4515	0.7592	0.4498	**0.7505**	0.2669	0.2723	**0.6505**	**0.6744**	0.0338	0.2671	0.5965	0.1214	0.2774	0.0141	0.0748	0.0126
3	0.8312	0.9627	0.8304	0.5556	0.4120	0.4043	**0.6505**	**0.6744**	0.0338	**0.4568**	**0.9444**	0.2143	0.4211	**0.4902**	0.1311	0.0266
4	0.9133	0.9808	0.9124	0.3915	**0.7080**	**0.4604**	0.6207	0.6588	**0.4828**	0.2500	0.8947	**0.3529**	**0.6383**	0.4324	**0.2189**	**0.0543**
5	**0.9161**	**0.9818**	**0.9152**	0.3798	0.3569	0.2806	0.5828	0.5195	0.0406	0.3136	**0.9444**	0.1405	0.1538	0.0000	0.0874	0.0229

Table 1: Scores achieved in the two Challenges

pute the ratio $p_{specific,general}(i)/p_{general}(i)$. For inferring location we adopted the per grid-cell language model based approach of [4]. It should be noted though that for clusters that contain geotagged images, we do not use the language models, but rather use the explicit coordinates to estimate the location.

3. EXPERIMENTS

3.1 Runs description in Challenge 1

In all runs of Challenge 1 we utilized a SVM classifier to learn the SEM. The following features were used to compute the input to the SEM for a pair of images: user (1 if both images have been uploaded by the same user, 0 otherwise), textual (title, tags and description, similarity computed using BM25 and cosine), taken and upload time, spatial (if available) and visual information (SURF descriptors aggregated using a VLAD scheme [8] as well as features extracted using Overfeat [7], a popular convolutional net, similarity for both is computed using Euclidean distance). In Run 1 we apply our basic approach, without using any visual features and we take the predictions of the SEM as they are, i.e. we do not change the classification threshold. In Run 2 we only add the visual features. In Run 3 we use the probabilities that are provided by the SVM classifier and set the threshold to 0.995, achieving the true positive and true negative rates that were mentioned earlier. In Run 4 we attempt to improve the results by increasing the set of candidate neighbours: after the graph has been constructed by predicting the SEM output for each image's candidate neighbours, we add to the candidate neighbours of each image the neighbours of its actual neighbours and predict the output of the SEM for them as well. In Run 5 we do not use blocking and compute the output of the SEM for all pairs of images.

3.2 Runs description in Challenge 2

In Run 1 of Challenge 2 we perform the classification by computing the ratio $p_{specific}(i)/p_{general}(i)$ and setting the threshold θ to 1. In Run 2, we perform the classification by computing the ratio $p_{specific,general}(i)/p_{general}(i)$ and again setting the threshold to 1. In Run 3 and Run 4 we use the models of Run 2 and Run 1 respectively, but with different threshold values per query. Each threshold is selected according to the evaluation results of the methodology in the corresponding development queries. For queries Test-9 and Test-10 where there are no analogous development queries, we used the maximum threshold from the other queries. In Runs 1 to 4 we perform classification per event, that is, we aggregate all images of an event and then perform the classification. In Run 5 on the other hand we perform classification per item and then perform the aggregation by majority vote. Also, in Run 5, the same approach in language models and threshold values as in Run 3 has been followed.

4. RESULTS AND DISCUSSION

4.1 Challenge 1

Table 1 shows the scores we achieved in Challenge 1. The main thing to note is that Runs 3, 4 and 5 that use the modified classification threshold show a very clear improvement over Runs 1 and 2 that do not. Moreover, it appears that appropriately expanding the candidate neighbours (Run 4 over Run 3) can also provide a significant improvement. Additionally, there is some further improvement in Run 5, that does not use blocking, over Run 4, but the improvement is very small. All in all, it can be said that strong blocking is useful in order to make the application of the method more scalable, but can lead to somewhat decreased performance.

4.2 Challenge 2

Table 1 shows the average scores that we achieved over all 10 queries of Challenge 2. We note that Run 3 and Run 4 give the best average scores meaning that the selected threshold has a significant influence in the accuracy of the classification results. Test queries perform better when a calibration of the threshold value comes first. The classification of an event by handling photos in cluster uniformly performs better than having an individual classification result per photo. It should also be mentioned that considering only queries that include location criteria, the performance is significantly higher. In particular, for those queries, in Run 4 we achieve an F-score of 0.6331.

5. ACKNOWLEDGMENTS

The work was supported by the European Commission under contract FP7-287975 SocialSensor.

6. REFERENCES

[1] D. Jurafsky and J. H. Martin. *Speech and Language Processing.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2000.

[2] G. Petkos, S. Papadopoulos, and Y. Kompatsiaris. Social event detection using multimodal clustering and integrating supervisory signals. In *Proc. of ICMR 2012.*

[3] G. Petkos, S. Papadopoulos, V. Mezaris, and Y. Kompatsiaris. Social event detection at MediaEval 2014: Challenges, datasets, and evaluation. In *Proceedings of the MediaEval 2014 Multimedia Benchmark Workshop*, 2014.

[4] A. Popescu. CEA list's participation at MediaEval 2013 Placing Task. In *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop*, 2013.

[5] T. Reuter and P. Cimiano. Event-based classification of social media streams. In *Proceedings of ICMR 2012.*

[6] M. Schinas, E. Mantziou, S. Papadopoulos, G. Petkos, and Y. Kompatsiaris. CERTH @ MediaEval 2013 Social Event Detection Task. In *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop*, 2013.

[7] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.

[8] E. Spyromitros-Xioufis, S. Papadopoulos, I. Kompatsiaris, G. Tsoumakas, and I. Vlahavas. An empirical study on the combination of SURF features with VLAD vectors for image search. WIAMIS, 2012.