Supervised Clustering of Social Media Streams

Martin Wistuba
University of Hildesheim
Information Systems & Machine Learning Lab
wistuba@ismll.de

Lars Schmidt-Thieme
University of Hildesheim
Information Systems & Machine Learning Lab
schmidt-thieme@ismll.de

ABSTRACT

In this paper we present our approach for the Social Event Detection Task 1 of the MediaEval 2013. We address the problem of event detection and clustering by learning a distance measure between two images in a supervised way. Then, we apply a variant of the Quality Threshold clustering to detect events and assign the images accordingly. We can show that the performance measures do not decrease for an increasing number of documents and report the results achieved for the challenge.

1. INTRODUCTION

This paper presents our approach to tackle Task 1 of the MediaEval Social Event Detection 2013 Challenge [7]. The task is to cluster images into an unknown number of events in such a way that they belong to each other. For the required run only meta information like title and description may be used whereas for the general runs more information can be considered. Here, we only discuss an approach for the required run.

2. FRAMEWORK DESCRIPTION

In this section we describe how our clustering approach works. Therefore, we first introduce the used features, explain our preprocessing process before we then define how we learn the similarity metric between two documents. Finally, our incremental clustering approach based on Quality Threshold clustering is explained.

2.1 Features

We represent a pair (d_i, d_j) of two documents d_i, d_j by a feature vector $x \in \mathbb{R}^m$ of m features. We have chosen the same nine features as Reuter et. al. [5]. Additionally, a further feature was used, indicating whether the document was created by the same user $(+1)$ or not (-1). If a feature cannot be computed because the information is missing, it is assumed to be 0.

MediaEval 2013 Workshop, October 18-19, 2013, Barcelona, Spain

2.2 Preprocessing

Textual information like title, tags and description is stemmed using a Porter Stemmer [3]. Additionally, the documents are sorted by the time of creation in ascending order. If the time of creation is unknown, the time of its upload is used instead.

2.3 Similarity Measure

Related work in this field [1, 6] prefer using SVMs to learn the similarity between two documents but for our clustering approach it has proven to be better to use Factorization Machines [4] instead. We randomly sampled 4,000 positive and 4,000 negative document pair examples. A document pair example (d_i, d_j) is positive if d_i and d_j belong to the same event, negative otherwise. The positive pairs were labeled with 1, the negative with 0. Then we trained the model of Factorization Machines (FM), i.e.

$$\hat{y}(x) = w_0 + \sum_{i=1}^{m} w_i x_i + \sum_{i=1}^{m} \sum_{j=i+1}^{m} v_i^T v_j x_i x_j$$

by using stochastic gradient descent. Here, w_0 is the global bias, w_i models the strength of the i-th variable and $v_i^T v_j$ models the interaction between the i-th and j-th variable where $V \in \mathbb{R}^{m \times k}$. As a hyperparameter search combined with those of the clustering would have been too time-intensive, we tuned the learning rate α and the regularization rate λ such that the root mean square error was acceptable. Concluding, we have chosen $\alpha = 0.05$, $\lambda = 0$ and $k = 1$. In the following section we will see that it is more important to choose the right hyperparameters for the clustering method.

2.4 Clustering Method

As the number of clusters is unknown and for application in practice, an incremental, threshold-based clustering technique is preferable as argued by Becker et. al. [1] we decided to use Quality Threshold clustering (QT) [2]. Because it is computationally intensive as much as $\mathcal{O}\left(n^3\right)$, an approximation was needed to speed it up. Previous work [1, 6] has used single-pass methods, but we were expecting better results by sticking to the QT idea. Instead of applying QT onto the full data, we split it into disjoint batches $b_1, \ldots, b_{\lceil n/l \rceil}$ of size l. Choosing l small enough makes it feasible to apply QT onto the batches. To also allow documents in the following batches to be placed into a cluster from documents in the previous batches, a representative of each cluster was kept. The representative of a cluster C is the document $d_{\mathcal{R}} = \arg\min_{d_i \in C} \sum_{d_j \in C} \delta(d_i, d_j)^2$, which is motivated by the smallest enclosing circle. Assuming that the represen-

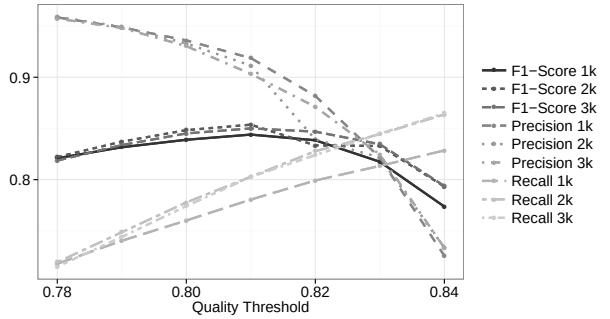

Figure 1: Excerpt of the grid search for batch sizes $l \in \{1000, 2000, 3000\}$ and quality threshold $\mu \in \{0.78, \ldots, 0.84\}$. Decreasing l and μ improves the precision, increasing them the recall.

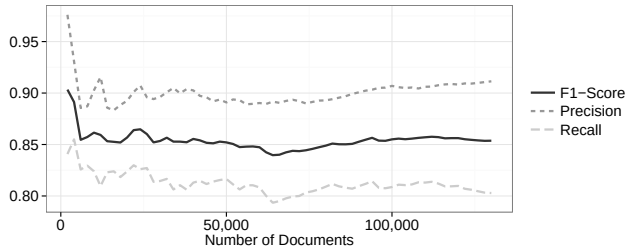

Figure 2: For the used validation set the evaluation scores seem to be stable for a growing number of documents. A threshold of $\mu = 0.81$ and a batch size of $l = 2,000$ was used.

tative is actually the center of the smallest enclosing circle, only documents with a distance of at most $\frac{\mu}{2}$ can be clustered to the same cluster for the following batches, where μ is the threshold.

3. EXPERIMENTS

For the clustering approach two hyperparameters are needed: the quality threshold μ and the batch size l. We estimated them using a grid search on 130,000 documents which is approximately the size of the testing set. The results identified that there is probably only one global optimum, but also that it is possible to trade precision with recall with only a small loss of the F_1-Score. For this challenge this is not of importance but as already stated by Reuter et. al. [5], a higher precision is more important for applications. A part of our grid search is shown in Figure 1. Finally, for the testing set we have chosen $\mu = 0.81$ and $l = 2,000$.

Another interesting fact of this approach is that it seems to be stable for a larger number of documents as shown in Figure 2. Reuter et. al. [5] has reported worse results for the algorithms presented by Becker et. al. and Reuter et. al. [1, 5] if the number of documents grow. Even though they have used a different dataset, a decrease in performance of the F_1-Score from around 87% for 10,000 documents to 74% for 100,000 documents cannot be neglected.

The final challenge results on the test set are presented in Table 1. The results are even better than those on the validation set. The reason for this is that the validation set was more complex as it contained more and smaller cluster. We recognized the larger clusters on the test set while computing the clusters. This led to really high clustering times for few batches such that we decided to stop clustering a batch if it took more than two hours computing time. The difference between the validation and test set also led to non-optimal hyperparameters as a threshold of $\mu = 0.82$ looks more promising.

Table 1: Final results on the test set for different hyperparameters.

Hyperparameter setting	F_1-Score	NMI
$\mu = 0.81$, $l = 2,000$	0.8720	0.9606
$\mu = 0.81$, $l = 1,000$	0.8712	0.9643
$\mu = 0.81$, $l = 1,500$	0.8755	0.9641
$\mu = 0.82$, $l = 1,500$	0.8784	0.9655

4. CONCLUSIONS

The presented algorithm promises to be a good method for this problem especially for bigger datasets. Therefore, a comparison to state of the art algorithms using the same dataset and features would be interesting. Possibly, blocking can also be applied to our approach to further improve the performance and especially the speed. As QT can be parallelized, this could be another possibility to improve the speed.

5. REFERENCES

[1] H. Becker, M. Naaman, and L. Gravano. Learning similarity metrics for event identification in social media. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 291–300, New York, NY, USA, 2010. ACM.

[2] L. J. Heyer, S. Kruglyak, and S. Yooseph. Exploring Expression Data: Identification and Analysis of Coexpressed Genes. *Genome Research*, 9(11):1106–1115, Nov. 1999.

[3] M. Porter. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3):130–137, 1980.

[4] S. Rendle. Factorization machines. In G. I. Webb, B. L. 0001, C. Zhang, D. Gunopulos, and X. Wu, editors, *ICDM*, pages 995–1000. IEEE Computer Society, 2010.

[5] T. Reuter and P. Cimiano. Event-based Classification of Social Media Streams. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, ICMR '12, pages 22:1–22:8, New York, NY, USA, 2012. ACM.

[6] T. Reuter, P. Cimiano, L. Drumond, K. Buza, and L. Schmidt-Thieme. Scalable event-based clustering of social media via record linkage techniques. In L. A. Adamic, R. A. Baeza-Yates, and S. Counts, editors, *ICWSM*. The AAAI Press, 2011.

[7] T. Reuter, S. Papadopoulos, V. Mezaris, P. Cimiano, C. de Vries, and S. Geva. Social Event Detection at MediaEval 2013: Challenges, datasets, and evaluation. In *MediaEval 2013 Workshop*, Barcelona, Spain, October 18-19 2013.