NTNU@MediaEval 2011 Social Event Detection Task (SED)

Massimiliano Ruocco and Heri Ramampiaro
Data and Information Management Group, Department of Computer and Information Science
Norwegian University of Science and Technology
Sem Saelands vei 7-9 NO-7491 Trondheim, Norway
{ruocco,heri}@idi.ntnu.no

ABSTRACT

In this paper we present the system used to solve the challenges of the *Social Event Detection* (SED) task at *MediaEval 2011* challenge.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

1. INTRODUCTION

This work is part of the MediaEval 2011 challenge. The general purpose of the challenge was to propose an event retrieval system. In particular, we proposed a system to solve two specific event extraction challenges. In the first challenge, the main purpose was to retrieve all *soccer events* in Rome and Barcelona and in the second challenge, we were asked to retrieve all events from two specified venues in Amsterdam (NL) and Barcelona (ES) within a certain temporal range. The results of the queries were presented as groups of images - i.e, one group per event. More specific details of the challenge can be found in [2].

2. SYSTEM OVERVIEW

In this section we give a detailed presentation of our proposed algorithm. Figure 1 shows an overview of our system.

2.1 Query Expansion

In a social event retrieval context, a query can be splitted and mapped in three different parts according to the general parameters characterizing an event: (1) *what*, i.e., which kind of event we are looking for, (2) *where*, i.e., the venue, name of place, city or region where the event that we are looking for takes place, (3) *when*, i.e., the time, interval when the event happens. In both challenges the *where* part of the query is expanded in the first block. For *Challenge 1* the *where* part is created with all the stadium names in Rome and Barcelona, in all languages while for *Challenge 2*, it was created with the names of the two venues specified in the challenge. For both challenges, the geographical location (latitude and longitude) are also extracted. In order to retrieve these information, a set of *SparQL* queries are submitted to *DBpedia*[1] database by using the *Jena*[2] interface for java. To be more specific, for *Challenge 1*, names in different languages and geographical information are extracted by selecting from the *DBpedia* category `Football_venues_in_Italy` the occurrences based on the city of Rome and Barcelona. For *Challenge 2*, the geographical location and names related to the requested venues were extracted using the *LastFM API*[3] and used as query into the *LastFM* database. The output of this block is a set of queries $Q = \{Q_1, ..., Q_N\}$, where each subset $Q_i = \{q_{i1}, ..., q_{iM}\}$ refers to all queries related to a venue and each $q_{ij} = \{\mathbf{T}, \mathbf{g}\}$ is composed of two different parts: a textual part with different names of the venues, and a spatial part with a pair of real numbers representing the latitude and longitude of the venues.

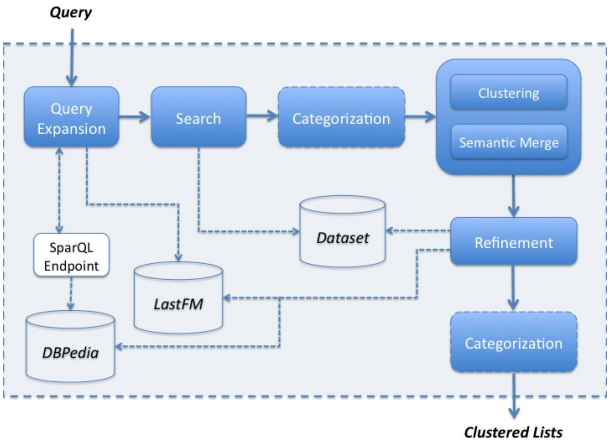

Figure 1: System Overview

2.2 Search

The queries are submitted to the search engine over the dataset. In our work, we use *Solr*[4] search engine to index the dataset and perform the search. The search is done as a

[1]http://dbpedia.org/
[2]http://jena.sourceforge.net/
[3]http://www.last.fm/api
[4]http://lucene.apache.org/solr/

MediaEval 2011 Workshop, September 1-2, 2011, Pisa, Italy

mix of spatial (by using latitude and longitude values) and textual search. The data are indexed based on the textual metadata, including `Title`, `Description` and `Tags`. The search is then performed over all the three different metadata. In particular for *Challenge 2* the queries are boolean queries with all the terms in AND, while in the *Challenge 1* these conditions are more relaxed and the terms of each query are composed with the boolean operator OR. The reason for this is that in this challenge, a categorizer is provided as next step to filter out non-relevant retrieved occurrences.

2.3 Categorization

The input of this block is a list of pictures with their metadata. This module is used only for *Challenge 1* to extract pictures related to a soccer event. The categorization is performed over the three textual metadata for each picture, i.e., `Title`, `Description` and `Tags`. The different runs will exploit the descriptivity of each kind of metadata in the categorization process (see Section 3). To categorize the pictures the *SemanticHacker API*[5] over the different textual metadata was used. The categories produced are based on the *Open Directory Project*[6]. The pictures are filtered by only keeping those categorized with a category that has radix `Sports/Soccer`.

2.4 Clustering and Merging

The previous block returns a set of filtered pictures (*Challenge 1*) related to soccer events or pictures taken in the venues specified in the search step (*Challenge 2*) and grouped based on the venues. In this step the temporal information will be used to group the temporal related pictures. In that way the resulting clusters are finally grouped according to their temporal and locational information. To perform the clustering process the *Quality Thresold Clustering* (QT) algorithm is used [1]. This algorithm does not require to specify in advance the number of clusters and even it is computationally expensive, it is used only on retrieved documents. The resulting clusters may be semantically related and belonging to the same event. To merge semantically similar clusters a graph is built, where the nodes of the graph are the clusters, and two nodes are connected if they share at least a tag representing a named entity of an event or of an artist. To extract the named entities, we use the tags and submit them as queries to *LastFM* for the artist names and *DBPedia* for event names. Clusters are merged by finding the connected component as in [3].

2.5 Refinement

The resulting clusters may be incomplete, i.e. the dataset may contain other pictures related to the event clusters extracted but not retrieved in the search step. The refinement module is used here to query the dataset by using the (1) top-k frequent tags and (2) top-k frequent entity names (artists and events). The results of the refinement step can still be filtered to avoid retrieving non-relevant occurrences.

3. EXPERIMENTS AND RESULTS

In this section, we present the different runs and their evaluation over different metrics. Table 1 provides a summary of our results.

[5]http://textwise.com/api
[6]http://www.dmoz.org/

3.1 Challenge 1

Two different runs were performed in the first challenge. In the first run (*Run 1*) all the workflow of the system is performed excluding the refinement step and semantic merge between clusters. For *Run 1* the categorization is performed by using only `Tag` metadata, while for the second run (*Run 2*), we also include the `Title` and `Description` metadata. From the results obtained (see Table 1) we can observe that including other metadata than tags resulted in a decrease of precision, probably due to the lack of descriptiveness of the other metadata.

3.2 Challenge 2

For this challenge, three different runs were performed. The *Run 1* (the baseline run) executed the algorithm without including the semantic merge and refinement steps. In both *Run 2* and *Run 3* the semantic merge and refinement steps were performed. Semantic merge was done by considering each cluster represented by the named entity representing events or artists. Moreover in *Run 2*, refinement is performed by querying the top-100 tags and the temporal range in which each cluster is closed. In *Run 3*, we used the entity names representing artists or events extracted from the set of tags of each cluster.

	Challenge 1		*Challenge 2*		
	Run 1	**Run 2**	**Run 1**	**Run 2**	**Run 3**
Precision	**94.26**	92.47	74.70	77.91	**78.85**
Recall	38.48	**43.16**	37.99	55.06	**56.83**
F-Measure	54.65	**58.65**	50.36	64.52	**66.05**
NMI	0.4613	**0.4752**	0.4101	0.5049	**0.6448**

Table 1: Evaluation measures of the runs

4. CONCLUSIONS

We have presented a system to extract events for the given two challenges. As described in this paper, the best result in terms of precision was obtained in the first challenge by using only the tags for the categorization step, while the other evaluation measure were better when using all the textual metadata. In the second challenge the best result was obtained using the complete workflow of the algorithm, i.e. using refinement step, in particular using the entity names in the refinement query for each cluster. Our future experiments, especially for the first challenge, will include the use of the refinement step and semantic merge over the totality of the results (instead of applying it over groups of results coming from the query).

5. REFERENCES

[1] 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.

[2] S. Papadopoulos, R. Troncy, V. Mezaris, B. Huet, and I. Kompatsiaris. Social Event Detection at MediaEval 2011: Challenges, Dataset and Evaluation. In *MediaEval 2011 Workshop*, Pisa, Italy, September 1-2 2011.

[3] M. Ruocco and H. Ramampiaro. Event clusters detection on flickr images using a suffix-tree structure. *Multimedia, International Symposium on*, 0:41–48, 2010.