UPMC at MediaEval 2016 Retrieving Diverse Social Images Task

Sabrina Tollari
Sorbonne Universités, UPMC Univ Paris 06, UMR CNRS 7606 LIP6, 75252 PARIS cedex 05, France
Sabrina.Tollari@lip6.fr

ABSTRACT

In the MediaEval 2016 Retrieving Diverse Social Images Task, we proposed a general framework based on agglomerative hierarchical clustering (AHC). We tested the provided credibility descriptors as a vector input for our AHC. The results on devset showed that this vector based on the credibility descriptors is the best feature, but unfortunately that is not confirmed on testset. To merge several features, we chose to merge feature similarities. Tests on devset showed that to merge similarities using linear or weighted-max operators gave, most of the time, better results than using only one feature. This results is partially confirmed on testset.

1. INTRODUCTION

Contrary to previous years, in 2016, the task [3] addresses the use case of a general ad-hoc image retrieval system. General cases are more difficult to tackle, because the system can't be adapted to a particular application. Another difference is the use of the `F1@20` metrics, which means that we are not only interested in diversity, but also to find a balance between relevance and diversity, that is more difficult to handle. In the task of 2013, we proposed a framework [4] which, first, tries to improve relevance and, then, makes a clustering to improve diversity. This strategy has obtained good results and can handle general cases. So this year, we use the same strategy, but we adapt the parameters to the use of `F1@20` metrics, i.e., not only to improve diversity, but to find a balance between relevance and diversity.

2. FRAMEWORK

For each query, we apply the following framework. Step 1 (optional): Re-rank Flickr baseline to improve relevance according to text features. Step 2: Cluster the N first results using Agglomerative Hierarchical Clustering (AHC). Step 3: Sort the images in each cluster using their rank in Step 1, sort the clusters according to the rank of the image on the top of each cluster. Step 4: Re-rank the results alternating images from different clusters.

The AHC [2] is a robust method that can handle different kind of features. Applying the AHC to query results provides a hierarchy of image clusters. In order to obtain groups of similar images, we cut the hierarchy to obtain a fixed number k of unordered clusters (see [5] for details).

MediaEval 2016 Workshop, October 20-21, 2016, Hilversum, Netherlands

The AHC needs a measure to compare two documents. A document can be describe by several features (text, visual, etc.). To take advantage of several features, we need a way to merge them. We choose to merge similarities. Some of the features are associated with a distance, others with a similarity. In order to have only similarities, all distances are transformed using the classical formula: let $\delta(x, y)$ be a distance between x and y, then the similarity is defined as: $\text{sim}(x, y) = 1/(1 + \delta(x, y))$.

Let f_1 and f_2 be two features and $\tau \in [0, 1]$, we compute a linear fusion of feature similarities by:

$$\text{sim}_{\texttt{Linear(f1,f2,}\tau\texttt{)}}(x, y) = \tau \cdot \text{sim}_{f_1}(x, y) + (1 - \tau) \cdot \text{sim}_{f_2}(x, y).$$

Let n be the number of features. Let's choose wisely a weight w_i for each feature f_i, such as $\sum_{i=1}^{n} w_i = 1$. We compute a weighted-max fusion similarities by:

$$\text{sim}_{\texttt{WMax(f1,w1,f2,w2,}\cdots\texttt{,fn,wn)}}(x, y) = \max_{i \in \{1,\cdots,n\}} w_i \cdot \text{sim}_{f_i}(x, y).$$

3. EXPERIMENTS AND RESULTS

Text re-ranking (Step 1).

Using vector space model (VSM) with tf-idf weights and cosinus similarity, we tested the choice of textual information fields (Title (`t`), Description (`d`), Tags (`t`), Username(`u`)). We also tested several stemmer. We notice no significant difference with or without stemmers, the reason may be because there are only a few words in the query title. So we choose not to use stemmer in all the experiments. Finally, its seems that globally `ttu` gives slightly better `P@20`.

Features for clustering in Step 2.

We tested several combinations of textual information fields. Finally, for text clustering, the best solution on `devset` is to use all the fields (`tdtu`) and a similarity based on the Euclidean distance. It seems that to use in addition the Description field tends to produce more diversity than using `ttu`, because documents are more dissimilar between them.

We tested the provided visual features `cnn_gen` and `cnn_ad`. On most of our experiments, it seems that `cnn_ad` gives slightly better or better results than `cnn_gen`. We also tested several features from the Lire library [1, 6]: the ScalableColor feature (`ScalCol`) — a histogram in HSV color space encoded by a Haar transform — gives the best results.

Using the provided credibility descriptors, we built, for each image, normalized real vectors of 13 dimensions (noted `cred`) (NaN, null and missing values — $\simeq 3.5\%$ of the credibility descriptor values — are replaced by random values).

Table 1: Run results. Between brackets, gain in percentage compared to the `devset` baseline or to the `testset` worst run. The number of documents for clustering per query is 300. k is the selected number of clusters

Run	Step 1	Steps 2-4: AHCCompl Features	k	devset P@20	devset CR@20	devset F1@20	testset P@20	testset CR@20	testset F1@20
baseline	-	-	-	0.698(ref.)	0.371(ref.)	0.467(ref.)	-	-	-
VSM	VSM(ttu)	-	-	0.772(+11%)	0.397(+7%)	0.507(+9%)	-	-	-
rand	VSM(ttu)	random features	50	0.771(+10%)	0.410(+11%)	0.522(+12%)	-	-	-
user	VSM(ttu)	username	50	0.761(+9%)	0.485(+31%)	0.578(+24%)	-	-	-
run 1	-	ScalCol	20	0.631(-10%)	0.432(+16%)	0.498(+7%)	0.520(ref.)	0.400(ref.)	0.430(ref.)
run 2	VSM(ttu)	tdtu	50	0.768(+10%)	0.471(+27%)	0.569(+22%)	0.697(+34%)	0.486(+22%)	0.552(+28%)
run 3	VSM(ttu)	Linear(tdtu, ScalCol,0.02)	50	0.767(+10%)	0.487(+31%)	0.582(+25%)	0.696(+34%)	**0.494**(+24%)	**0.553**(+29%)
run 4	VSM(ttu)	cred	50	0.767(+10%)	0.491(+32%)	0.585(+25%)	0.681(+31%)	0.487(+22%)	0.543(+26%)
run 5	VSM(ttu)	WMax(tdtu,0.014, ScalCol,0.97,cred,0.016)	50	0.771(+10%)	**0.493**(+33%)	**0.588**(+26%)	0.686(+32%)	0.487(+22%)	0.544(+26%)
Number of queries				70			64		

Clustering parameters in Step 2.

When varying features and parameters, we noticed that, on `devset`, globally, complete linkage (`AHCCompl`) gave better results than single or average linkages.

For each query, 300 results were provided. Usually, there are more relevant documents in the first results than in the end of the result list. Is it worth for the system to take time to cluster online 300 results in order to improve the `F1@20` of the first 20 documents ? We made several experiments varying the diversity methods, parameters, features and number of input documents. Globally, we didn't get a lot of differences in term of better `F1@20` between 150, 200, 250 or 300 documents. The only real difference depends on the number of clusters. Usually, the more documents are in the input set, the more the number of clusters should be high to obtain goods results: around 20 clusters for 150 documents, and around 50 clusters for 300 documents. Finally, we choose to take 300 documents because the peak of the curve is wider than with 150 documents.

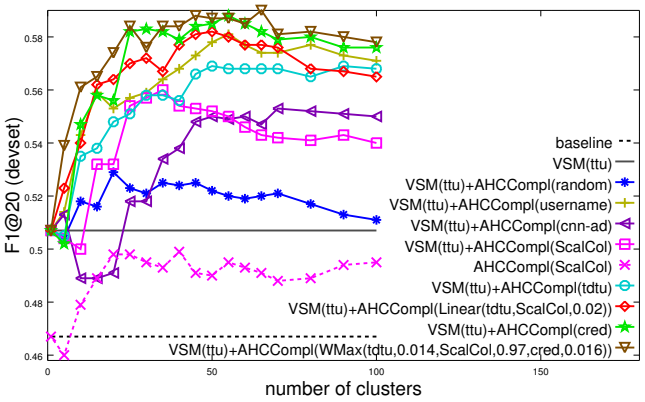

Figure 1: Some of the results on `devset` varying the number of clusters (300 documents per query)

Reference runs and run results.

The `baseline` run is the FlickR ranking. `VSM(ttu)` run is obtained using VSM on `ttu` fields and without clustering. To have some comparison elements, we also tested: a clustering of random features (documents are represented by vectors of 5 random values) and a clustering using only the `username` (two documents with same `username` are similars).

As the queries are only composed of text, we cannot apply a Step 1 to improve relevance in the case of `run 1` (visual only). Figure 1 shows that `AHCCompl(ScalCol)` (clustering on `ScalCol` features without Step 1) gives lower results than `VSM(ttu)+AHCCompl(ScalCol)` (with Step 1), but in both cases, visual features give lower results than `tdtu` or `cred` features.

The best number of clusters to use is always an open question. If we want the best `CR@20`, most of the time is better to take 20 clusters, unfortunately with 20 clusters the `P@20` is often the worst. So, in order to optimise `F1@20` and according to the curves on `devset` (see Figure 1), we choose for `run 2` to `run 5` to take 50 clusters. This choice seems to give a good compromise between relevance and diversity.

On `devset`, best results using only one feature is obtained using `cred`. The reason may be because, in the case of `cred`, images with the same userid have the same vectors, so these images will be in the same cluster and these images are often about the same subtopic. In Figure 1, we can notice that the clustering on `username` gives better results than on text only (`tdtu`) or visual only (`ScalCol`), but lower results than on `cred`. So there must be also another reason. If some images have similar credibility descriptors, that means that their users have the same characteristics. But it is not clear why these characteristics are interesting for diversity. To try to show that `cred` is a good feature for diversity whatever the diversity method, we tried this feature with a greedy algorithm and we obtained the same conclusions (on `devset`). Unfortunately, on `testset`, the text only run (`run 2`) gives better results than `cred` one (`run 4`) (see Table 1). So this result cannot be generalized and may depend of `devset`.

As the visual similarity are not normalized, we needed to carefully optimize the weights of the linear and of the weighted-max fusion operators. On `devset`, the weighted-max fusion using `tdtu`, `ScalCol` and `cred` gave the best results for all our experiments. But as `cred` is not so good on `testset`, `run 5` does not give very good results. Finally, the linear fusion between text (`tdtu`) and visual (`ScalCol`) gives the best results on `testset` (`run 3`).

Despite the fact that we use different kind of features, the `F1D@20` for `run 2` to `run 5` are very close (from 0.543 to 0.553) that means it's difficult to make reliable conclusion on the best feature or on the interest of similarity fusion.

4. REFERENCES

[1] `http://www.semanticmetadata.net/lire`.

[2] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, Inc., p. 552, 2000.

[3] B. Ionescu, A. L. Gînscă, M. Zaharieva, B. Boteanu, M. Lupu, and H. Müller. Retrieving diverse social images at MediaEval 2016: Challenge, dataset and evaluation. In *MediaEval 2016 Workshop*, Hilversum, Netherlands, October 20-21 2016.

[4] C. Kuoman, S. Tollari, and M. Detyniecki. UPMC at MediaEval2013: Relevance by text and diversity by visual clustering. In *MediaEval 2013 Workshop*, 2013.

[5] C. Kuoman, S. Tollari, and M. Detyniecki. Using tree of concepts and hierarchical reordering for diversity in image retrieval. In *CBMI*, pages 251–256, 2013.

[6] M. Lux and S. A. Chatzichristofis. Lire: Lucene image retrieval: An extensible java CBIR library. In *ACM International Conference on Multimedia*, pages 1085–1088, 2008.