Combining text/image in WikipediaMM task 2009

Christophe Moulin, Cécile Barat, Cédric Lemaître, Mathias Géry, Christophe Ducottet, Christine Largeron

Combining text/image in WikipediaMM task 2009

Christophe Moulin, Cécile Barat, Cédric Lemaître, Mathias Géry, Christophe Ducottet, Christine Largeron

Université de Lyon, F-69003, Lyon, France
Université de Saint-Étienne, F-42000, Saint-Étienne, France
CNRS UMR5516, Laboratoire Hubert Curien
{christophe.moulin, cecile.barat, cedric.lemaitre, mathias.gery, ducottet,largeron}@univ-st-etienne.fr

Abstract. This paper reports our multimedia information retrieval experiments carried out for the ImageCLEF track 2009. In 2008, we proposed a multimedia document model defined as a vector of textual and visual terms weighted using a tf.idf approch [5]. For our second participation, our goal was to improve this previous model in the following ways: 1) use of additional information for the textual part (legend and image bounding text extracted from the original documents, 2) use of different image detectors and descriptors, 3) new text / image combination approach. Results allow to evaluate the benefits of these different improvements.

1 Introduction

ImageCLEFwiki is a multimedia collection where each document is composed of text and one image. User needs are represented by queries ("topics"), which are also multimedia. Therefore, a multimedia document model is necessary to handle such a collection. In 2008, we proposed a first model that combines text and image information for multimedia retrieval [5]. This year, we improve our model adding textual information and using different detectors and descriptors for the visual information. Moreover we use a linear combination to merge our textual and visual results. After presenting our model, we will explain the submitted runs and the obtained results. We will finish by introducing our future work.

2 Visual and textual document model

The document model we defined for ImageCLEF 2008 lets us rank documents depending on the query using different methods. Firstly, we explain the key features of our approach to rank documents according to a query using only textual information. Secondly, we describe how we extend the method to handle the visual information. Finally, we present our method for combining textual and visual results.

2.1 Textual representation model

As in the vector space model introducted by Salton et al. [7], we represent a document d_i as a vector of weights $w_{i,j}$. Each $w_{i,j}$ corresponds to the importance of the term t_j in the document d_i computed by multiplying $tf_{i,j}$ and idf_j, where $tf_{i,j}$ is the term frequency that characterizes the frequency of the term t_j in the document d_i. The idf_j is the inverse document frequency that quantifies the importance of the term t_j over the corpus of documents. $w_{i,j}$ is high when the term t_j is frequent in the document d_i but rare in the others. We use $tf_{i,j}$ and idf_j defined in the Okapi formula by Robertson et al [6] by :

$$tf_{i,j} = \frac{k_1 n_{i,j}}{n_{i,j} + k_2(1 - b + b\frac{|d_i|}{d_{avg}})}$$

where $n_{i,j}$ is the occurrence of the term t_j in the document d_i, $|d_i|$ the size of the document d_i and d_{avg} the average size of all documents in the corpus and k_1, k_2 and b are three constants.

$$idf_j = \log \frac{|D| - |\{d_i | u_j \in d_i\}| + 0.5}{|\{d_i | t_j \in d_i\}| + 0.5}$$

where $|D|$ is the size of the corpus and $|\{d_i | t_j \in d_i\}|$ the number of documents where the term t_j occurs at least one time.

If we consider a query q_k as a short document, we can represent it as a vector of weights. A score is then computed between the query q_k and a document d_i as shown in table 1. The main difference between $score^1$ and $score^2$ is the representation of the query. In the first score, the weight of the query is defined by its tf only while in the second score the weight is equal to $tf.idf$. Note that for $tf_{k,j}$, $b = 0$ because $|d_k|$ and $|d_{avg}|$ are not defined for a query.

Scoring	Parameters $tf_{i,j}$	Parameters $tf_{k,j}$
$score^1(q_k, d_i) = \sum_{t_j \in q_k} tf_{i,j} idf_j tf_{k,j}$	$k_1 = 2.2$ $k_2 = 1.2$ $b = 0.75$	$k_1 = 8$ $k_2 = 7$ $b = 0$
$score^2(q_k, d_i) = \sum_{t_j \in q_k} tf_{i,j} idf_j tf_{k,j} idf_j$	$k_1 = 1$ $k_2 = 1$ $b = 0.5$	$k_1 = 1$ $k_2 = 1$ $b = 0$

Table 1. Scoring equations and their default parameters[8].

Different sources of text are available. The legend provided with images is often very short and sometimes useless: for example, when the text deals with the copyright of the image or when it gives details about the user who uploaded the image. In order to gain information, we aim at using the original text extracted

from the wikipedia documents in which images appear. We consider a text fragment aroung the image. The size of the window is tuned using wikipediaMM 2008 collection as a training collection. We add this text to the legend of the image and we index both the added text and the original legend. The indexing is performed with the Lemur software[8].

2.2 Visual representation model

In order to combine the visual information with the textual one, we also represent images as a vector of weights. Provided we are able to extract visual words from images, it is possible to use the $tf.idf$ formula in the same way as in the textual model. It is therefore necessary to create a visual vocabulary $V = \{v_1, ..., v_j, ..., v_{|V|}\}$ as in [2]. For that purpose, we use 3 different descriptors. The first one ($meanstd$) is the same as in [5]. Each image is partitioned into 16x16 cells. Each cell is described by a six dimensional vector which corresponds to the mean and the standard deviation of $\frac{R}{R+G+B}$, $\frac{G}{R+G+B}$ and $\frac{R+G+B}{3*255}$ where R, G and B are the red, green and blue components of the cell. The second (named $sift_1$) and third (named $sift_2$) descriptors are based on the well known sift descriptor [3]. The $sift_1$ firstly detects regions of interest using the MSER method as in [4] while the $sift_2$ one uses a regular partitioning as in the $meanstd$ descriptor.

For each of our 3 descriptors, we apply a k-means algorithm [1] to obtain a vocabulary of 10'000 visual terms. Each visual term represents a cluster of feature vectors.

Then, each image can be represented using a vector of visual terms. Local features are first calculated using one of the 3 descriptors. Then visual terms are determined by seeking, for each feature, the closest visual term (according to the euclidian distance) in the corresponding visual vocabulary.

In the same way as for textual words, the weight of each visual term is computed using a $tf.idf$ approach.

2.3 Combination

In order to combine textual and visual results we use two different methods. The first one is a simple intersection between the results obtained with the textual query and with the visual one. The second one corresponds to a linear combination between the textual and the visual scores.

$$score(q_k, d_i) = \alpha score_V(q_k, d_i) + (1 - \alpha) score_T(q_k, d_i)$$

The α parameter lets us add more or less visual information. We calculate its optimal value using the queries from the ImageCLEFwiki 2008 track as a training set.

3 Experiments

Using the model described in the previous section, we present runs submitted to ImageCLEFwiki and the results we obtained.

3.1 Submitted runs

run_id	run	score	text	image	combination
$LaHC\_1$	LaHC_TXT_okapi	$score^1$	*legend*	-	-
$LaHC\_2$	LaHC_TXT_tfidf	$score^2$	*legend*	-	-
$LaHC\_3$	run_inter_TXT_IMG_Meanstd	$score^2$	*legend*	*meanstd*	intersection
$LaHC\_4$	run_inter_TXT_IMG_Sift	$score^2$	*legend*	$sift_1$	intersection
$LaHC\_5$	run_TXTIMG_Meanstd_0.015	$score^2$	*legend*	*meanstd*	α= 0.015
$LaHC\_6$	run_TXTIMG_Meanstd_0.025	$score^2$	*legend*	*meanstd*	α= 0.025
$LaHC\_7$	run_TXTIMG_Sift_0.012	$score^2$	*legend*	$sift_1$	α= 0.012
$LaHC\_8$	run_inter_TXT_IMG_Siftdense	$score^2$	*legend*	$sift_2$	*intersection*
$LaHC\_9$	run_TXT_100_3_1_5	$score^2$	100 *char*	*legend*	-
$LaHC\_10$	run_TXT_50_3_1_5	$score^2$	50 *char*	*legend*	-
$LaHC\_11$	run_TXTIMG_100_3_1_5_meanstd	$score^2$	100 *char*	*meanstd*	α= 0.025
$LaHC\_12$	run_TXTIMG_50_3_1_5_meanstd	$score^2$	50 *char*	*meanstd*	α= 0.025
$LaHC\_13$	run_TXTIMG_Siftdense_0.084	$score^2$	*legend*	$sift_2$	α= 0.084

- *meanstd*: regular partitioning + color descriptor
- $sift_1$: MSER detector + sift descriptor
- $sift_2$: regular partitioning + sift descriptor
- *legend*: text of the image document
- n *char*: size (n characters) of the text window around the image in the original wikipedia documents

Table 2. Presentation of the runs

All the runs are entirely automatic and are summarized on table 2. We define a baseline, $LaHC\_1$, that corresponds to a pure text model. It uses only textual terms for the query and scoring of documents. We calculate the $score^1$ for each image using terms of the textual content. The image name or bounding characters are not considered. We do not use neither feedback nor query expansion. Since $score^1$ is applied, the query terms are weighted with their frequency tf.

Using only the text, we perform 3 other runs: the $LaHC\_2$ is the same as the baseline except that the query is represented by its $tf.idf$ rather than its tf. The $LaHC\_9$ and $LaHC\_10$ are two other text only runs that make use of the bounding text around the image in the original wikipedia document. The $LaHC\_9$ adds 100 characters before and after the image while the $LaHC\_10$ adds 50 characters.

All other runs exploit both the textual and the visual information of documents. The $LaHC\_3$, $LaHC\_4$ and $LaHC\_8$ are obtained after an intersection of the text only query results ($LaHC\_2$) and the image query using the *meanstd*, the $sift_1$ and the $sift_2$ descriptors. The other runs are obtained from a linear combination of the textual and the visual scores. $LaHC\_5$, $LaHC\_6$, $LaHC\_7$ and $LaHC\_13$ use the textual scores of $LaHC\_2$ and the visual scores of a visual descriptor (*meanstd*, $sift_1$ and $sift_2$). $LaHC\_11$ and $LaHC\_12$ combine the textual scores of $LaHC\_9$ and $LaHC\_10$ with the visual scores of the *meanstd* descriptor.

4 Results

All the obtained results are summarized in Table 3. On the whole results, our team ranks 2nd on 8 participants[1]. As we can see, the best results are obtained when we combine the image bounding text and the *meanstd* descriptor. We could have obtained better results if we had combined the image bounding text and the $sift_2$ descriptor. Indeed, comparing results of text-image runs $LaHC\_6$, $LaHC\_7$ and $LaHC\_13$, we can notice that the best visual descriptor is $sift_2$, followed by *meanstd* and $sift_1$. The last three results obtained after an intersection are the worst results, but if we compare them in term of precision, they are the best ones. Indeed, one document over six is relevant. As a rule, combining the textual information with the visual one always improves the results and return more relevant documents which is very encouraging.

rank	run	map	num_ret	num_rel_ret
5	run_TXTIMG_100_3_1_5_meanstd	0.2178	44993	1213
6	run_TXTIMG_50_3_1_5_meanstd	0.2148	44993	1218
14	run_TXTIMG_Siftdense_0.084	0.1903	44993	1212
15	run_TXT_100_3_1_5	0.1890	38004	1205
16	run_TXT_50_3_1_5	0.1880	37041	1198
20	run_TXTIMG_Meanstd_0.025	0.1845	44993	1208
21	run_TXTIMG_Sift_0.012	0.1807	44995	1200
24	run_TXTIMG_Meanstd_0.015	0.1792	44993	1213
33	LaHC_TXT_tfidf	0.1667	35611	1192
44	LaHC_TXT_okapi	0.1432	35611	1164
52	run_inter_TXT_IMG_Siftdense	0.0365	619	142
53	run_inter_TXT_IMG_Meanstd	0.0338	574	76
54	run_inter_TXT_IMG_Sift	0.0321	637	120

Table 3. Presentation of the results

[1] `http://imageclef.org/2009/wikiMM-results`

5 Conclusion

In this article we proposed improvements to our multimedia model we introduced in [5]. The first one was to use image bounding text extracted from the original documents, the second was to use sift based image descriptors for the visual part and the third one was to add a text/image combination approach. A series of thirteen runs was submitted using the ImageCLEFwiki 2009 collection. The first analysis of the results allowed to make the three following remarks. It's better to use the image bounding text than the legend only. The sift descriptor is better than our previous color descriptor provided it is calculated on a regular partitioning. The text-image combination is a winning strategy which can be implemented by linear combination of textual and visual scores.

For future work, we aim to combine the textual information with more than just one visual descriptor. Moreover as the visual information importance depends on the query, we also plan to learn a different α parameter for each query.

6 Acknowledgements

This work was partly supported by the LIMA project[2] and the Web Intelligence project[3] which are 2 Rhône-Alpes region projects of ISLE cluster[4].

References

1. Halil Bisgin. Parallel clustering algorithms with application to climatology. Technical report, Informatics Institute, Istanbul Technical University, Turkey., 2008.
2. Frédéric Jurie and Bill Triggs. Creating efficient codebooks for visual recognition. In International Conference on Computer Vision, 2005.
3. David G. Lowe. Object recognition from local scale-invariant features. In Proceedings of the International Conference on Computer Vision ICCV, Corfu, pages 1150–1157, 1999.
4. K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. International Journal of Computer Vision, 65(1-2):43–72, 2005.
5. Christophe Moulin, Cecile Barat, Mathias Gry, Christophe Ducottet, and Christine Largeron. Ujm at imageclefwiki 2008. In ImageCLEF 2008, 2008.
6. Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Aarron Gull, and Marianna Lau. Okapi at trec-3. In Text REtrieval Conference, pages 21–30, 1994.
7. Gerard Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. Communations of the ACM, 18(11):613–620, 1975.
8. C. Zhai. Notes on the lemur tfidf model. Technical report, Carnegie Mellon University, *http://www.lemurproject.com*, 2001.

[2] LIMA project: `http://liris.cnrs.fr/lima/`
[3] WI project: `http://www.web-intelligence-rhone-alpes.org/`
[4] ISLE cluster: `http://ksup-gu.grenet.fr/isle`