Multimodal information approaches for the Wikipedia collection at ImageCLEF 2011

R. Granados[1], J. Benavent[2], X. Benavent[2], E. de Ves[2], Ana García-Serrano[1]

[1] Universidad Nacional de Educación a Distancia, UNED
[2] Universitat de Valencia
rgranados@lsi.uned.es, xaro.benavent@uv.es, agarcia@lsi.uned.es,
esther.deves@uv.es, jobego@alumni.uv.es

Abstract. The main goal of this paper it is to present our experiments in ImageCLEF 2011 Campaign (Wikipedia retrieval task). This edition we focused on applying different strategies of merging multimodal information, textual and visual, following both early and late fusion approaches. Our best runs are in the top ten of the global list, at positions 8, 9 and 10 with MAP 0.3405, 0.3367 and 0.323, being the second best group of the contest. Moreover, 18 of the 20 runs submitted are above the average MAP of its own modality (textual or mixed). In our system, the TBIR module works firstly and acts as a filter, and the CBIR system works only with the filtered sub-collection. The two ranked lists are fused using its own probability in a final ranked list. The best run of the TBIR system is in position 14 with a MAP of 0.3044, and uses subsystems IDRA and Lucene, fusing monolingual experiments carried out with IDRA preprocessing and Lucene search engine, taking into account extra information from Wikipedia articles. The best result at the CBIR system is obtained by using a logistic regression relevance feedback algorithm and CEDD low-level features.

Keywords: Information Retrieval, Textual-based Retrieval, Content-Based Image Retrieval, Relevance feedback, Merge Results Lists, Indexing, Multimedia, Multimodal, Fusion

1 Introduction

The UNED-UV is a research group with researchers from two different universities in Spain, the Universidad Nacional de Educación a Distancia (UNED) and the Valencia University (UV). This research group is working together [1] [2] [7] since ImageCLEF08 edition.

Two kinds of experiments were submitted to the 2011 Wikipedia Retrieval edition [3]: pure textual runs (TBIR), and mixed (with visual, CBIR). For textual experiments different approaches (stemming, use of articles info, and named entities recognition) were tested in order to evaluate the differences among them. For the mixed runs, as in 2010 presented ones [7], the TBIR system works firstly over the whole database as a filter and then the CBIR only works over the filtered collection. Finally, the fusion

module gets a ranked list, merging the textual and visual lists taking into account the probabilities obtained by each of the modules individually. The merging module for the multimodal information is the main goal of study of our group for this edition.

The TBIR subsystem includes the UNED own implemented tool IDRA (InDexing and Retrieving Automatically) [4] which includes several functionalities: text extraction and preprocessing, indexation following a Vector Space Model (VSM) approach using TF-IDF weighted vectors, retrieval based on the cosine function, a connection to a basic Lucene [12] configuration, and some merges utilities. The CBIR subsystem uses its own low-level features or the CEDD ones [11], depending on the experiment in order to test the influence of the low-level features in the final results. Two different algorithms have also been used: a logistic regression relevance feedback algorithm and an automatic algorithm with the Tanimoto distance. A more detailed presentation of the system, the submitted experiments, and the obtained results are included in the following sections.

2 System Description

To carry out the Wikipedia retrieval task, it has been used a three modules architecture, as shown at Fig. 1. The figure illustrates the global system, which includes the TBIR (text based image retrieval) module, the CBIR (content based image retrieval) module, and the fusion one.

As the conceptual meaning of a topic is initially better captured by the text module itself than by the visual one, the textual module works first as a filter for the visual one, which works only with the sub-collection filtered by the textual module. Each module gets a ranked list based on a similarity score or probability (the TBIR module uses the textual information to obtain these scores while the CBIR one uses the visual one). From now on, we call textual probability (Pt) to the probability given by the textual module and image probability (Pi) to the probability given by the visual module. The way of merging these two probabilities is studied at the fusion module.

The TBIR subsystem is based on the IDRA tool, which allows to preprocess the textual information associated with the images in the collection, and to index and retrieve using both its own implemented search engine (based on a VSM approach), and a basic configuration of Lucene [12].

The CBIR subsystem uses its own low-level features or the CEDD features depending on the experiment (in order to test the influence of the kind of low-level features in the final result), and its own logistic regression relevance feedback algorithm. Also, an automatic algorithm, which uses the Tanimoto distance as the score for ranking images in the collection, has been implemented in order to compare the performance of this distance with the logistic regression algorithm.

Each of the two subsystems, TBIR and CBIR, generates a ranked list with a certain probability and this multimodal information is merged at the fusion module. Different ways of merging this information is tested. Moreover, merging algorithms are used inside the TBIR subsystem to fuse different textual result lists from monolingual experiments in order to obtain multilingual results, as other fusing techniques are used

inside the CBIR subsystem. All details of the different levels of merging information and the algorithms used are explained in the following sections.

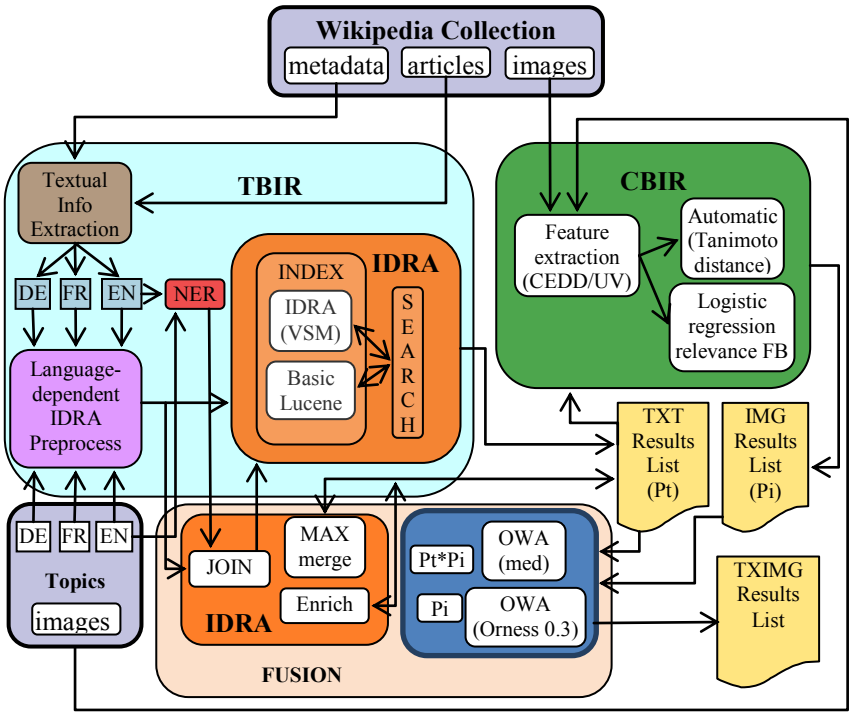

Fig. 1. System overview.

2.1 Text-based Index and Image Retrieval (TBIR)

This module is in charge of the index and search of the images in the collection, based on the textual information in the metadata files associated with each of these images. IDRA tool is used to extract, select and preprocess this information. Additional resources, as STILUS [9] or Snowball [10], are required in the preprocessing step. Finally, both IDRA and Lucene search engines are used for the index and retrieval of the preprocessed textual info, obtaining a ranked results list with the retrieved relevant images for each query.

The components shown in Fig. 1 within the TBIR subsystem are the followings:

Textual Info Extraction. Two different textual information sources can be differentiated in the collection: the metadata files and the articles files. The metadata XML tags extracted, using the JDOM Java API, are <name> and the general <comment> for all languages, and <description>, <comment> and <caption> for each particular language (English, French and Dutch).

The <caption> tag from metadata files may include a link to the article/s from Wikipedia where images appear. This information is taken into account in some of the experiments, extracting the title and the categories from the linked articles.

The final output of this component will be the selected textual information describing the images, coming from both the metadata and the articles, and separated depending on the language: text (EN), text (FR) and text (DE).

Language-dependent IDRA Preprocess. This component processes the selected text in three steps: 1) special characters deletion: characters with no statistical meaning, like punctuation marks or accents, are eliminated; 2) stopwords detection: exclusion of semantic empty words from specifics lists for each language; and 3) stemming: for reducing inflected or derived words to their stem, base or root form. A different algorithm is needed to perform stemming for each one of the languages. Stemmers from Snowball [10] are used in the experimentation.

NER (STILUS). The Named Entities Recognition is carried out by the 'List Entities' functionality of the STILUS-Core API [9]. Different forms of the detected entities (from general and variants) are took into account and considered as textual information in the corresponding experiments.

Index&Search. Once completed extraction and preprocess, both IDRA tool and Lucene will be used to index the selected text, and to retrieve relevant images for the proposed queries.

IDRA indexation is based on the VSM approach using TF-IDF (term frequency – inverse document frequency) weighted vectors. This approach consists in calculating the weights vectors for each one of the images selected texts. Each vector is compounded by the TF-IDF weights values of the different words in the collection. TF-IDF weight is a statistical measure used to evaluate how important a word is to a text in a concrete collection. These weights are normalized using the Euclidean distance.

IDRA search will launch the textual queries from the topics (English, French or Dutch) against a concrete index, obtaining this way the corresponding "TXT Results List". For each one of the queries, IDRA calculates its corresponding weights vector in the same way as in the index. Then, the similarity between the query and an image text will depend on the proximity of their associated vectors calculated by the cosine measure. This similarity value will be calculated between the query and all the images associated text indexed. Then images are ranked in descending order of relevance in the "TXT Results List".

Lucene indexation and search can be executed from IDRA tool. Selected texts already preprocessed with IDRA are indexed with Lucene following a basic implementation that uses the WhiteSpaceAnalyzer which just separates tokens and doesn't apply any other linguistic preprocess.

2.2 Content-Based Information and Visual Retrieval

The VISION-Team at the Computer Science Department of the University of Valencia has its own CBIR system, and that has been used in previous ImageCLEF

editions since our first participation in 2008. Last edition, the focus of the work was in testing three different visual algorithms applied to the results retrieved by the text module: the automatic, the relevance feedback and the query expansion obtaining the best results with the relevance feedback algorithm. Therefore, this edition we have used the relevance feedback algorithm and the work has been focus on testing the behavior of our own low-level features with the low-level features given by the organization (the CEDD algorithm described in [11]).

Extraction of low level features. As in most CBIR systems, a feature vector represents each image. The first step at the Visual Retrieval system is extracting these features for all the images on the database and for each image in the query topic. We use different low-level features describing color and texture to build a vector of 293 components based on color and texture information.

- **Color information**: Color information has been extracted calculating both local and global histograms of the images using 10x3 bins on the HS color system. Local histograms have been calculated dividing the images in four fragments of the same size. Therefore, a feature vector of 222 components represents the color information of the image.
- **Texture information**: Two types of texture features are computed: the granulometric distribution function, using the coefficients that result of fitting the distribution function with a B-spline basis. And, the Spatial Size Distribution. We have used two different versions of it by using as the structuring elements for the morphological operation that get size both a horizontal and a vertical segment [1]. This gives us a texture feature vector of 71 components.

We assume that the conceptual meaning of a question is better captured by the text module than by a visual module when they work individually. Therefore, the task of the visual module is to re-rank the textual result list taking into account the information of the query images given at each topic.

Automatic algorithm. This is a classical algorithm in a CBIR system. Each image in the database has an associated low level feature vector. Concretely, we have used for this algorithm the low level features given by the organization (CEDD).

The second step is to calculate the similarity measurement between the feature vectors of each image on the database and the N query images. The distance metric applied in our experiments is the Tanimoto. As we have N query images, we will obtain N visual result lists, one for each query image in the topic. These N result lists are merged by using an average OWA operator.

Relevance feedback algorithm based on logistic regression. This algorithm works differently to the two previous ones. Therefore, we will explain the concept of relevance feedback and the adjustments made to get a good performance of the algorithm for the proposed tasks [5]. Relevance feedback is a term used to describe the actions performed by a user to interactively improve the results of a query by reformulating it. An initial query formulated by a user may not fully capture his/her wishes. Users then typically change the query manually and re-execute the search until they are satisfied. By using relevance feedback, the system learns a new query

that better captures the user's need for information. The user enters his/her preferences at each iteration through the selection of relevant and non-relevant images.

We will explain the way the logistic regression relevance feedback algorithm works. Let us consider the (random) variable Y giving the user evaluation where Y=1 means that the image is positively evaluated and Y=0 means a negative evaluation. Each image in the database has been previously described by using low-level features in such a way that the j-th image has the k-dimensional feature vector xj associated. Our data will consist of (xj, yj), with j=1,…,n, where n is the total number of images, xj is the feature vector and yj the user evaluation (1=positive and 0=negative). The image feature vector x is known for any image and we intend to predict the associated value of Y. In this work, we have used a logistic regression where P(Y=1|x) i.e. the probability that Y=1 (the user evaluates the image positively) given the feature vector x, is related with the systematic part of the model (a linear combination of the feature vector) by means of the logit function. For a binary response variable Y and p explanatory variables X1,…,Xp, the model for $\pi(x)=P(Y=1|x)$ at values $x=(x1,\ldots,xp)$ of predictors is $\text{logit}[\pi(x)]=\alpha+\beta 1x1+\ldots+\beta pxp$, where $\text{logit}[\pi(x)]=\ln(\pi(x)/(1-\pi(x)))$. The model parameters are obtained by maximizing the likelihood function given by:

$$l(\beta)=\prod^{n}\pi(x_i)^{y_i}[1-\pi(x_i)]^{1-y_i} \qquad (\mathbf{1})$$

The maximum likelihood estimators (MLE) of the parameter vector β are calculated by using an iterative method.

We have a major difficulty when having to adjust a global regression model in which we take the whole set of variables into account, because the number of selected images (the number of positive plus negative images) is typically smaller than the number of characteristics. In this case, the regression model adjusted has as many parameters as the number of data and many relevant variables could be not considered. In order to solve this problem, our proposal is to adjust different smaller regression models: each model considers only a subset of variables consisting of semantically related characteristics of the image. Consequently, each sub-model will associate a different relevance probability to a given image x, and we face the question of how to combine them in order to rank the database according to the user's preferences. This problem has been solved by means of an ordered averaged weighted operator (OWA) [6].

In our case, we have adapted the manual relevance feedback to an automatic performance. The examples and the counter-examples (positive and negative images) are automatically selected for each topic. The examples are the query images of the topic plus N images taken from the first positions of the textual result list. The M counter-examples are obtained by applying a procedure which chooses J random images from the whole database (without images in the textual list). This J images are ranked by the Euclidean distance and the latest M images are taken as negative examples.

2.3 Multimodal Fusion

Different types of fusion are needed in different steps of the experimentation. Depending on the point the fusion is carried out, it is called early fusion (at feature level) or late fusion (at decision level) [8]. Moreover, fusion can be applied among resources from different modes (text and image), or the same (different sources of text). Late fusion algorithms are used when fusing multiple modalities in the semantic space, at decision level, and when fusing textual results from monolingual experiments. All mixed runs submitted fuse together textual and visual runs at this level, combining the mono-modal decisions (results lists) from each modality.

2.3.1 Fusion in textual runs

Several fusion approaches are used within the textual module, following both early and late fusion techniques. Fusion at feature level (early fusion) is used when combining text in different languages [8], or from different processes (i.e. NER). A late fusion approach is used when fusing together the decisions from each of the monolingual processes.

The different implemented algorithms, and the purposes they were built for, are explained bellow:

JOIN. Early fusion approach which just concatenates several lists of terms coming from different sources, obtaining only one. The level of fusion of this method is feature (early fusion), as it merges the components of the representation vectors of each image.

This approach is used in two kinds of experiments: 1) to merge the terms from the different languages (EN, FR, DE) describing each image in the collection into a unique multilingual representation (run9); and 2) to fuse at feature level the terms coming from the metadata textual information, with those obtained from the NER process (run10).

MAXmerge. Used to fuse together different results lists. This algorithm is included in IDRA tool and, for each query, selects the results from the different lists which have a higher relevance/similarity value, independently of the list in which the results appear in. The merging of the results corresponds to a decision level fusion (late fusion), where individual decisions working with each language are mixed in a unique results list, which is multilingual. (run4, run8)

Enrich. Late fusion algorithm used to merge two results lists, also included in IDRA. The algorithm fuses together a main list with a support one. If a retrieved image appears in both lists for the same query, the relevance of this result in the merged list will be increased in the following way (normalized from 0 to 1):

$$new\,\mathrm{Re}\,l = main\,\mathrm{Re}\,l + \frac{\sup \mathrm{Re}\,l}{pos\,\mathrm{Re}\,l + 1} \qquad \textbf{(2)}$$

newRel: relevance value in the merged list
supRel: relevance value in the support list
mainRel: relevance value in the main list
posRel: position in the support list

Every results appearing in the support list but not in the main one (for each query), will be added at the end of the results for each query. In this case, relevance values will be normalized according with the lower value in this moment.

This method is used (run7) to enrich the results of one of the English monolingual experiments with those from a NE-based experiment. The improvement (or not) of these fusion will be appreciable just in those queries (a total of 9) where named entities where detected.

2.3.2 Fusion in the Visual Module

There are two points where a fusion algorithm is needed inside the visual procedures.

The first one is in relevance feedback algorithm because, as we have explained before, our proposal is to adjust different smaller regression models. Consequently, each sub-model will associate a different relevance probability to a given image x. An ordered averaged weighted operator (OWA) with an orness of 0.5 has been used for this purpose, as explained in section 2.2.

On the other hand, the automatic algorithm generates N result visual lists depending on the number of query images. These N lists are merged in one result final list by using the Mathematical aggregation operators OWA with an orness of 0.5.

2.3.3 Multimodal Fusion. Merging textual and visual lists

The late fusion module is focused on merging the two probabilities obtained for each of the images from the textual and from the visual module independently. Different ways of merging these two probabilities have been tested:

- Pi. Using only the image probability to rank the filtered textual list. In this experiment the textual probability is used only on the first step to filter a sub-collection of the most similar images to the topic, and then the re-ranking is only based on the visual information.
- Pi*Pt: The ranking is made by using the score computed by the product of the two probabilities. At these experiments both textual and visual information is used to obtain the final re-ranked list.The OWA operator has been in the late fusion process too. This operator transforms a finite number of inputs into a single output (in our case the inputs are the Pi and Pt probabilities). With the OWA operator no weight is associated with any particular input; instead, the relative magnitude of the input decides which weight corresponds to each input. The aggregation weights used for these experiments are the weights which correspond to an orness with values 0.3 (this means that a weight of 0.3 is given to the higher probability value) and 0.5 (this is like an average operator).

3 Experiments (submitted runs)

A total of 20 runs were finally submitted to the task, 10 text-based and 10 mixed combining both textual and visual techniques. A schematic description of all the 20 runs is available in the followings tables.

Table 1. Submitted textual experiments.

ID	Lang	Details					
		System	md	stem	Art	Fusion	NER
run1	EN	IDRA	✓	-	-	-	-
run2	EN	IDRA	✓	✓	-	-	-
run3	EN	IDRA	✓	✓	✓	-	-
run4	ALL	IDRA	✓	✓	✓	MAXmerge (3en, 3fr, 3de)	-
run5	EN	IDRA+Lucene	✓	✓	✓	-	-
run6	FR	IDRA+ Lucene	✓	✓	✓	-	-
run7	EN	IDRA+ Lucene	✓	✓	✓	Enrich (5,NEs)	✓
run8	ALL	IDRA+ Lucene	✓	✓	✓	MAXmerge (5en, 5fr, 5de)	-
run 9	ALL	IDRA+ Lucene	✓	✓	✓	JOIN	-
run10	EN	IDRA+ Lucene	✓	✓	✓	JOIN	✓

Experiments 1, 2, 3 and 4 are fully run using the IDRA tool (pre-processing, indexing and retrieval). These runs try to compare different configuration possibilities: 1) applying stemming or not; 2) adding the articles textual information; 3) using of the textual information from the languages different from English (FR, DE). The stemming process for the different languages was carried out using Snowball.

The way we take into account the textual information in the related articles provided with the collection consists on extracting the title and the Wikipedia categories of the corresponding article. Run 4 uses metadata, stemming and articles (as run 3 in English) independently for each one of the 3 languages, and applies a late fusion algorithm (MAXmerge) in order to merge the obtained monolingual results.

Runs 5 and 6 use the same configuration as run 3, but using IDRA just to pre-process the data, and Lucene for index and search. Run 8 also uses IDRA and Lucene, and follows the same late fusion approach among monolingual experiments carried out with IDRA pre-processing and Lucene search engine. Run 9 fuses together the text from different languages at feature level, following an early fusion method (JOIN).

Runs 7 and 10 add the use of a named entities recognisor based in the STILUS-Core API. This information is mixed with the textual info already available in two ways: 1) concatenating the identified entities for each image with its own associated

text, before indexing, that is, merging at feature level (the same concatenation is done with the text of the queries); 2) searching independently using only entities for indexation and search (just in 9 queries with named entities detected), and using these results to enrich (late fusion algorithm) the results list from run 5 based in english textual information.

Table 2. Submitted mixed experiments.

ID	TXT run	Details			
		Low-level Features	Relevance feedback	Distance	Fusion
run11	run9	UV	✓	-	Pi
run12	run9	CEDD	✓	-	Pi
run13	run9	CEDD	Automatic	Tanimoto	OWA(Med)
run14	run9	CEDD	✓	-	Pt*Pi
run15	run9	CEDD	✓	-	OWA(Med)
run16	run9	CEDD	✓	-	OWA(Orness0.3)
run17	run9	UV	✓	-	Pt*Pi
run18	run8	CEDD	✓	-	Pt*Pi
run19	run8	CEDD	✓	-	OWA(Med)
run20	run8	CEDD	✓	-	OWA(Orness0.3)

Two textual based algorithms have been used for testing the different ways of merging the textual and the visual information, named as the run8 and the run9 (see table 1 for textual detail algorithms). Therefore, the block of runs 14, 15 and 16 is designed to be compared with runs 18, 19 and 20 in order to see the influence of the textual based algorithm used. In each of the group, we have test three ways of merging the textual and the visual information: the product of the textual and visual probabilities, an aggregation OWA operator using an ORNESS(0.3) or ORNESS(0.5) as weights.

Runs 11 and 12 have been made to test the influence of the low-level features used (UV or the CEDD), and also to test the fusion algorithm that only uses the visual information, image probability, to re-rank the final list. Finally, run13 is designed to test the two different vision algorithms used: the relevance feedback and the automatic algorithm with the Tanimoto distance.

4 Results

After the evaluation by the task organizers, our results for each of the submitted experiments are presented in Table 2. The table shows how our best results are for the mixed runs 18, 20 and 19 (at positions 8, 9 and 10 of the global result list, this is at the 10% first results). For the text modality, the best result is run 8 at position 14 (at the 15% first results). It is worth pointing out, that all our runs except two of them are

above the average of each modality (textual and mixed runs), and that for group classification we are the second group in the global result list.

Table 2. Results for the submitted experiments

Po	Run	Mode	MAP	P@10	P@20	R-prec.	Bpref
93	run1	Textual	0.1727	0.3040	0.2380	0.2140	0.1786
75	run2	Textual	0.2056	0.3700	0.2900	0.2518	0.2097
63	run3	Textual	0.2243	0.3980	0.3270	0.2702	0.2287
51	run4	Textual	0.2489	0.3800	0.3290	0.2913	0.2450
45	run5	Textual	0.2601	0.4560	0.3670	0.3014	0.2600
98	run6	Textual	0.1561	0.3580	0.2600	0.2111	0.1813
50	run7	Textual	0.2515	0.4460	0.3660	0.2997	0.2541
14	run8	Textual	**0.3044**	**0.5060**	**0.4040**	**0.3435**	**0.3012**
36	run9	Textual	0.2758	0.4520	0.3550	0.3154	0.2771
55	run10	Textual	0.2403	0.4520	0.3510	0.2908	0.2458
107	run11	Mixed	0.0553	0.1180	0.1030	0.0816	0.0631
108	run12	Mixed	0.0516	0.0880	0.0950	0.0802	0.0579
21	run13	Mixed	0.2869	0.5040	0.4060	0.3306	0.2909
15	run14	Mixed	0.3006	0.5200	0.4030	0.3379	0.2983
20	run15	Mixed	0.2869	0.5040	0.4060	0.3306	0.2909
17	run16	Mixed	0.2980	0.5000	0.4030	0.3338	0.2954
16	run17	Mixed	0.3006	0.4960	0.3960	0.3376	0.2996
8	run18	Mixed	**0.3405**	0.5420	**0.4500**	**0.3752**	**0.3378**
10	run19	Mixed	0.3233	0.5400	0.4230	0.3586	0.3217
9	run20	Mixed	0.3367	**0.5460**	0.4410	0.3673	0.3314
Average		Textual	0.2169	0.3973	0.3228	0.2668	0.2246
Best (pos11)		Textual	0.3141	0.5160	0.4270	0.3504	0.3107
Average		Mixed	0.2558	0.4542	0.3678	0.3049	0.2648
Best (pos 1)		Mixed	0.3880	0.6320	0.5100	0.4162	0.3847

Textual runs 1, 2 and 3 show how the application of stemming and the use of the textual information coming from the articles categories from Wikipedia have a positive influence in the final image retrieval. Analyzing results from runs 7 and 10, it can be observed that the recognition of named entities is not a very useful in terms of MAP, may be because only 9 of the queries contain any entity (results per topic should be detailed analyzed). Only two runs (from XRCE) have obtained better results than our best one, and these both use query expansion or feedback techniques.

Regarding to the textual fusion, it can be observed in our two best textual runs how late fusion (run8, 0.3044) obtains better results than early fusion (run9, 0.2758). The only difference between these two runs is that run9 fuses together the textual information from the three languages at the beginning of the process (at feature level), while run8 works independently with each language and combines the results at the end of the process (at decision level).

With respect to the mixed runs, our best result is run18 with a MAP of 0.3405, ranked in position number 8. This is also our best global result. This experiment uses the images filtered by run8 (our best textual result), low level features given by the

organization (CEDD), and the regression algorithm. The merging is performed by ranking the final list with probabilities Pi*Pt. Runs number 11 and 12, with MAP of 0.0553 and 0.0516, are our worst global results (position 107 and 108 of the global ranked MAP list). These two runs use only the image probability Pi for ranking the final result list.

5 Concluding Remarks and Future Work

Our best results are for the mixed modality, and they are at the position 8, 9 and 10, this is at the top ten of the contest. Moreover, 18 of the 20 runs submitted have their MAP above the average of its own modality (textual or mixed). These results mean that our strategy of fusing multimodal information as the textual and visual algorithms is on the top of the retrieval information strategies.

The best mixed run mentioned uses at the textual module IDRA and Lucene with fusion approach among monolingual experiments, and the visual module uses CEDD features as low-level features and the logistic regression relevance feedback algorithm; and fuses the two lists, textual and visual multiplying both probabilities. Our group will continue working tuning the two modules independently, in order to improve the results obtained.

In the textual modality we have discover the positive influence of taking into account the textual information extracted from Wikipedia categories, and the better performance when fusing together the textual information from different languages at decision level (late fusion, run8) than doing it at feature level (early fusion, run9).

Regarding the merging strategies, the combination of the multimodal information (textual and visual) at the decision level gives always better results than using only textual or visual information individually (runs 18,19 and 20 against run8 or run12). Among the different merging strategies presented, the best results are always obtained multiplying both probabilities, followed by the aggregation OWA operator at different values (runs 18, 20 and 19 respectively).

Acknowledgments. This work has been partially supported by projects BUSCAMEDIA (CEN-20091026), MA2VICMR (S2009/TIC-1542) and project MCYT TEC2009-12980

References

1. Ana García-Serrano, Xaro Benavent, Ruben Granados, José Miguel Goñi-Menoyo. Some results using different approaches to merge visual and text-based features in CLEF'08 photo collection. Lecture Notes in Computer Science, Evaluating Systems for Multilingual and Multimodal Information Access. Vol.: 5706/2009 Págs, 568-571. ISSN: 0302-9743.
2. Ana García-Serrano, Xaro Benavent, Ruben Granados, Esther de Ves, Jose Miguel Goñi. Multimedia Retrieval by Means of Merge of Results from Textual and Content Based Retrieval Subsystems. Lecture Notes in Computer Science. Multilingual Information Access Evaluation II. Multimedia Experiments. Revised selected papers from 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, September 30 - October 2,

2009. Volume 6242/2010. DOI: 10.1007/978-3-642-15751-6. ISBN: 978-3-642-15750-9. pp: 142-149. #pp: 8.
3. Theodora Tsikrika, Adrian Popescu, Jana Kludas. Overview of the wikipedia image retrieval task at ImageCLEF 2011. CLEF 2011 working notes, Amsterdam, The Netherlands, 2011.
4. Ruben Granados Muñoz, Ana García Serrano, José M. Goñi Menoyo. La herramienta IDRA (Indexing and Retrieving Automatically). Procesamiento del Lenguaje Natural, nº 43, Septiembre de 2009. XXV Conferencia de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN'09). San Sebastián, 2009.
5. Leon, T., Zuccarello, P., Ayala, G., de Ves, E., Domingo, J.: Applying logistic regression to relevance feedback in image retrieval systems, Pattern Recognition, vol. 40, pp. 2621--2632. (2007).
6. R. Yager. On ordered weighted averaging aggregation operators in multi criteria decision making. IEEE Transactions Systems Man and Cybernetics (1988). Vol. 18 pages 183-190.
7. Joan Benavent, Xaro Benavent, Esther de Ves, Ruben Granados, Ana García-Serrano. Experiences at ImageCLEF 2010 using CBIR and TBIR Mixing Information Approaches. CLEF 2010 LABs and Workshops, Notebook Papers, 22-23 September 2010, Padua, Italy. 2010. ISBN: 978-88-904810-0-0.
8. P. Atrey, M.A. Hossain, A.E. Saddik and M. Kankanhalli, Multimodal fusion for multimedia analysis: a survey, Multimedia Systems 16 (2010), pp. 345–379.
9. STILUS-Core de Daedalus. http://www.daedalus.es/productos/stilus/stilus-core.html
10. Snowball Stemmer. http://snowball.tartarus.org/
11. S. A. Chatzichristofis, K. Zagoris, Y. S. Boutalis and N. Papamarkos, "Accurate Image Retrieval based on Compact Composite Descriptors and Relevance Feedback Information", International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), Volume 24, Number 2 / February, 2010, pp. 207-244, World Scientific.
12. Apache Lucene. http://lucene.apache.org