

---

# Categorization in Multiple Category Systems

---

**Jean-Michel Renders**

**Eric Gaussier**

Xerox Research Centre Europe, 6, chemin de Maupertuis, 38420 Meylan, France

**Cyril Goutte**

Institute for Information Technology, National Research Council of Canada, 101, rue St-Jean-Bosco, Gatineau, QC K1A 0R6, Canada

**Francois Pacull**

**Gabriela Csurka**

Xerox Research Centre Europe, 6, chemin de Maupertuis, 38420 Meylan, France

JEAN-MICHEL.RENDERS@XRCE.XEROX.COM

ERIC.GAUSSIER@XRCE.XEROX.COM

CYRIL.GOUTTE@CNRC-NRC.GOV.CA

FRANCOIS.PACULL@XRCE.XEROX.COM

GABRIELA.CSURKA@XRCE.XEROX.COM

## Abstract

We explore the situation in which documents have to be categorized into more than one category system, a situation we refer to as *multiple-view categorization*. More particularly, we address the case where two different categorizers have already been built based on non-necessarily identical training sets, each one labeled using one category system. On the top of these categorizers considered as black-boxes, we propose some algorithms able to exploit a third training set containing a few examples annotated in both category systems. Such a situation arises for example in large companies where incoming mails have to be routed to several departments, each one relying on its own category system. We focus here on exploiting possible dependencies between category systems in order to refine the categorization decisions made by categorizers trained independently on different category systems. After a description of the multiple categorization problem, we present several possible solutions, based either on a categorization or reweighting approach, and compare them on real data. Lastly, we show how the multimedia categorization problem can be cast as a multiple categorization problem and assess our methods in this framework.

## 1. Multiple-View Categorization

Consider a situation where objects  $x$  must be filed in different category systems. This is typical, for example, for companies that use an internal document classification system in addition to an industry-wide classification scheme (eg the International Patent Classification, IPC<sup>1</sup>). The category systems may be independent, or, more likely, there may be some dependencies that we can exploit to improve the classification accuracy in each system.

Moreover, consider the case where, for historical, organizational or other practical reasons, one has already built categorizers for each category system, based on generally different training sets. Indeed, in order to fully exploit available resources, it is preferable not to restrict oneself to training data that have been annotated in both category systems (the intersection set). For instance, in the example above, it makes sense to train an internal document categoriser on all available internal documents, rather than internal documents that are also patents (ie in the IPC). It is quite frequent that categorizers for the different views may have been developed at different times and be already available. Finally, there is also a more principled reason : The alternative of retraining a single categorizer targeting all combinations of categories from the two systems is not practical, because the number of parameters explodes.

In our work, we assume that we already have independently trained classifiers for each category system (or view). There are operational reasons for this: In the same vein, multimedia document categorization

---

Appearing in *Proceedings of the 23<sup>rd</sup> International Conference on Machine Learning*, Pittsburgh, PA, 2006. Copyright 2006 by the author(s)/owner(s).

<sup>1</sup><http://www.wipo.int/classifications/ipc/ipc8/>

typically raises the same issue, namely the fact that there exist powerful mono-media categorizers based on mono-media features and category systems, but these categorizers used large, mono-media training data. Generic visual categorizers for scene categories such as those we will use in the application described hereafter, need several hundreds or sometimes thousands of training images; categories are quite broad and have to cope with very large variability (size, illumination, ...). The Dmoz corpus<sup>2</sup>, on the other hand, constitutes an incredible training data set for generic textual categorizers and it would be a pity that it could not be exploited. This kind of situation also occurs for an individual user who is archiving and organizing a family multimedia corpus: she/he has not enough training material to directly train a mixed-media categorizer and may want to exploit some standard mono-media categorizers.

We will now focus on the situation where we have *two* category systems. The extension to more category systems is straightforward, although the complexity usually increases as the product of the number of categories.

As the different category systems provide different ways to describe, or label, the data, we call this problem *multiple-view categorization*. Multiple-view categorization is related to, but significantly different from, multi-label categorization (Sebastiani, 2002). In multi-label categorization, an object may be assigned several labels from a given category system. In fact, finding out how many categories should be assigned to a given observation is often a significant issue. By contrast, in *multiple-view categorization*, we are essentially doing single label categorization in each of a plurality of category systems. We have to pick *one* label in each category system.

This is also different from the topic of “learning with multiple views”, which addresses the problem of learning from data where observations are represented by multiple independent sets of features. A typical example is a webpage represented either by its content or by the content of anchor text pointing to this page, or a multimedia document represented either by its text or its images. Co-training (Blum & Mitchell, 1998) is a typical example of such approaches. *Multiple-view categorization*, on the other hand, means that each document may be filed in different category systems. The categorizers used for each category system may in fact use the same features, or they may not (eg text and images).

<sup>2</sup><http://www.dmoz.org>

To formalize the problem somewhat, we assume that we have objects  $x$  and two category systems  $c_1 = \{1, \dots, K_1\}$  and  $c_2 = \{1, \dots, K_2\}$ . We have learned independent categorizers for both category systems,  $\mathbf{f}_1(x) = \hat{P}(c_1|x)$  and  $\mathbf{f}_2(x) = \hat{P}(c_2|x)$ . We further assume that these categorizers output a probability, either by rescaling or by calibration. The problem that we address is the following: given dependencies between the two category systems  $c_1$  and  $c_2$ , can we correct the scores  $\mathbf{f}_1(x)$  and  $\mathbf{f}_2(x)$  in order to improve the categorization accuracy?

In the next section, we briefly discuss the approach which consists in building an additional classification layer on top of the two independent categorizers. By analogy with the technique used for model combination (Wolpert, 1992), we call these “stacking approaches”. The main problem with these is that they may not scale up to large category systems. In the following section, we focus on reweighting approaches, and present two original reweighting schemes for correcting assignment probabilities in multiple-view categorization. We then present experimental results obtained first on a text categorization task with multiple category systems. We also present a multimedia categorization problem which we cast in terms of multiple-view categorization and we assess our methods in this framework.

## 2. Stacking Approaches

Assuming that, for practical as well as computational reasons, we do not wish to retrain a single model targeting all combinations of categories from both systems, the most obvious way to address the problem from a Machine Learning point of view is to build an additional layer on top of the independent categorizers. The outputs of the categorizers,  $\mathbf{f}_1(x)$  and  $\mathbf{f}_2(x)$ , are the input of this layer, and the final output of this additional layer is the relevant category system (or both category systems if needed). We call this the *stacking approach* by analogy with Wolpert’s stacked generalisation (Wolpert, 1992). Note however that the usual stacking combines the output of several classifiers for the *same* category systems, while in our framework, we combine the inputs of a single categorizer from each category system.

There are essentially two alternatives for the output layer. The first is to consider all pairs of categories from both systems. This is not practical as soon as the number of categories grow, as the number of parameters grows at least as  $K_1.K_2.(K_1 + K_2)$  (for a simple linear model). The second possibility is to divide the output layer in two classifier: one for the first cate-

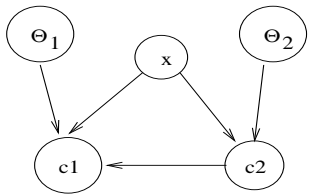


Figure 1. Graphical model for the asymmetric reweighting approach (model A).

gory system (with  $K_1$  outputs) and one for the second system (with  $K_2$  outputs). The number of parameters is therefore limited. For two linear models, we have about  $(K_1 + K_2)^2$  parameters.

We experiment with this approach using logistic regression in the output layer, ie a simple linear model with a *softmax* output. One drawback of the stacking approach is that the number of parameters still grows quite fast with the number of categories, especially when one set of categories is large. In addition, it requires an additional training step to train the output classifier(s). This contrasts with some of the reweighting approaches presented below, where the parameters have closed form expressions and no optimisation step is required.

### 3. Reweighting Approaches

#### 3.1. Model A

We first adopt two asymmetric reweighting models which capture potential correlations between categories pertaining to different category systems in an asymmetric way. Figure 1 displays the graphical model corresponding to the model reweighting the categories from  $c_1$ . A similar model is used for reweighting the categories in  $c_2$ .

From the model described in figure 1, the quantity we are interested in can be rewritten as:

$$\begin{aligned} P(c_1 = i|x, \theta_1, \theta_2) &= \sum_j P(c_1 = i, c_2 = j|x, \theta_1, \theta_2) \\ &= \sum_j P(c_2 = j|x, \theta_2) \times \\ &\quad P(c_1 = i|c_2 = j, x, \theta_1). \end{aligned}$$

We assume here that the quantity  $P(c_1 = i|c_2 = j, x, \theta_1)$  arises from two distinct contributions:  $P(c_1|x, \theta_1)$  and a positive potential  $\psi$  defined over  $c_1$  and  $c_2$ . Using this decomposition in the above equation, with appropriate normalization, and setting  $P(c_1 = i|x, \theta_1)$  (resp.  $P(c_2 = j|x, \theta_2)$ ) to  $\widehat{P}(c_1 = i|x, \theta_1)$  (resp.  $\widehat{P}(c_2 = j|x, \theta_2)$ ), we arrive at:

$$\begin{aligned} P(c_1 = i|x, \theta_1, \theta_2) &\approx \sum_j \widehat{P}(c_2 = j|x, \theta_2) \times \\ &\quad \frac{\psi(c_1 = i, c_2 = j) \widehat{P}(c_1 = i|x, \theta_1)}{\sum_k \psi(c_1 = k, c_2 = j) \widehat{P}(c_1 = k|x, \theta_1)} \end{aligned} \quad (1)$$

and similarly for  $c_2$ . The only values that need be estimated in eq. 1 are the values for  $\psi$ , which reflect the correlations between categories from the two category systems. As the decision function given by 1 is defined up to a scaling factor which does not depend on the value of  $c_1$ , we can impose, without loss of generality, the following constraints on  $\psi$ :  $\forall j, \sum_i \psi(c_1 = i, c_2 = j) = 1$ . We can then resort to a maximum likelihood approach, under the preceding constraints, to derive estimates for  $\psi$  from training examples (indexed by  $r$ ):

$$\begin{cases} \operatorname{argmax}_{\psi} \sum_r \log P(x^r, c_1^r, c_2^r | \theta_1, \theta_2) \\ (= \operatorname{argmax}_{\psi} \sum_r \log \frac{\psi(c_1^r, c_2^r) \widehat{P}(c_1^r | x^r, \theta_1)}{\sum_k \psi(k, c_2^r) \widehat{P}(k | x^r, \theta_1)}) \\ \text{under the constraints } \forall j, \sum_i \psi(i, j) = 1 \end{cases} \quad (2)$$

Interestingly, one can note that the (equivalent) reparametrization  $\psi_{ij} = \frac{e^{\beta_{ij}}}{\sum_i e^{\beta_{ij}}}$  leads to a concave, unconstrained maximization problem. The maximum is thus unique and can be found using standard optimization procedures.

Alternatively, one can resort to approximating  $\psi(i, j)$  by the empirical joint distribution  $\widehat{P}(i, j) = \#\{i, j\}/N$ , where  $\#\{i, j\}$  is the number of examples with labels  $c_1 = i$  and  $c_2 = j$  in the training set. In practice, we saw no significant difference between these two estimation procedures, and the results we report in section 4 were obtained with the latter one.

Once the  $\psi$  values have been estimated on the training set, the category decision for some new object  $x$  is simply:

$$c_1 = \operatorname{argmax}_i P(c_1 = i|x, \theta_1, \theta_2), \quad (3)$$

with  $P(c_1 = i|x, \theta_1, \theta_2)$  given by equation 1.

#### 3.2. Model B

We propose a symmetric reweighting model based on correcting the joint assignments of  $x$  in category pairs  $(c_1, c_2)$ . The correction relies on the knowledge of the joint category distribution  $P(c_1, c_2)$ . In practice, this distribution is not known beforehand but estimated on the data using, eg, Maximum Likelihood (as above). We first introduce additional category variables  $C^1$

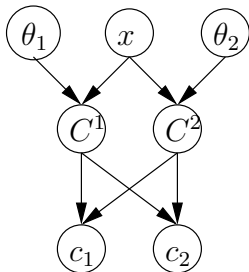


Figure 2. Graphical model for the symmetric reweighting formula (model B).

and  $C^2$  which represent the categories assigned by the independent categorizers  $\mathbf{f}_1$  and  $\mathbf{f}_2$ . The corresponding graphical model is in figure 2.

Conditional distributions for  $C^1$  and  $C^2$  are given by the independent categorizers:  $P(C^1|x, \theta_1) = \mathbf{f}_1(x)$  and  $P(C^2|x, \theta_2) = \mathbf{f}_2(x)$ . From now on we will keep the conditioning on  $\theta_1$  and/or  $\theta_2$  implicit. In order to correct the posterior assignment probabilities given by the independent models,  $P(C^1, C^2|x) = P(C^1|x)P(C^2|x)$ , let us consider the posterior probabilities for  $(c_1, c_2)$  and  $(C^1, C^2)$ :

$$\begin{aligned} P(c_1, c_2|x) &\propto P(x|c_1, c_2)P(c_1, c_2) \\ P(C^1, C^2|x) &\propto P(x|C^1, C^2)P(C^1, C^2) \end{aligned}$$

As the proportionality factor  $P(x)$  is identical for both, and  $P(C^1, C^2|x) = P(C^1|x)P(C^2|x)$ , we combine these into:

$$P(c_1, c_2|x) = \frac{P(x|c_1, c_2)P(c_1, c_2)}{P(x|C^1, C^2)P(C^1, C^2)} P(C^1|x)P(C^2|x)$$

The assumption we make is that for the relevant<sup>3</sup>classes,  $P(x|c_1, c_2)/P(x|C^1, C^2) \approx 1$ . As a consequence, the only parameters needed to obtain  $P(c_1, c_2|x)$  are  $P(c_1, c_2)$  and  $P(C^1, C^2)$ . They are estimated using the training set sample  $\mathcal{D} = \{x^r, c_1^r, c_2^r\}_{r=1 \dots N}$ . For  $P(c_1, c_2)$  we use the empirical estimate  $\hat{P}(c_1, c_2) = \#\{c_1, c_2\}/N$ , where  $\#\{c_1, c_2\}$  is the number of examples with labels  $c_1$  and  $c_2$  in  $\mathcal{D}$ . Note that this is the Maximum Likelihood estimator for the joint probability.

As  $P(C^1, C^2) = \int_x P(C^1, C^2|x)P(x)$ , we obtain a plug-in estimator using the empirical density  $\hat{P}(x) = \frac{1}{N} \sum_r \delta(x = x^r)$  as an estimator of  $P(x)$  ( $\delta(x = a)$  is the Dirac distribution in  $a$ ). Plugging this estimator

<sup>3</sup>This is likely to be very wrong when  $P(x|c_1, c_2)$  and  $P(x|C^1, C^2)$  are very small, but these are typically the classes for which it does not matter as the final decision focuses on classes with the highest probabilities.

in the above integral yields:

$$\hat{P}(C^1, C^2) = \frac{1}{N} \sum_r P(C^1|x^r)P(C^2|x^r). \quad (4)$$

The final formula for model B is:

$$\begin{aligned} P(c_1 = i, c_2 = j|x) &\propto \frac{\hat{P}(c_1 = i, c_2 = j)}{\hat{P}(C^1 = i, C^2 = j)} \\ &\times P(C^1 = i|x)P(C^2 = j|x) \end{aligned} \quad (5)$$

Marginalising  $c_2$  we get the following decision:

$$\begin{aligned} \hat{c}_1 &= \operatorname{argmax}_i P(C^1 = i|x) \times \\ &\sum_{c_2} \frac{\hat{P}(c_1 = i, c_2 = j)}{\hat{P}(C^1 = i, C^2 = j)} P(C^2 = j|x) \end{aligned} \quad (6)$$

and similarly for  $c_2$ .

Note that using weights  $\gamma(i, j) = \frac{\hat{P}(c_1 = i, c_2 = j)}{\hat{P}(C^1 = i, C^2 = j)}$  this in fact amounts to correcting the assignment probabilities with a simple reweighting formula:

$$P(c_1 = i|x) = P(C^1 = i|x) \sum_j \gamma(i, j) P(C^2 = j|x) \quad (7)$$

(with a similar expression for  $P(c_2 = i|x)$ ).

## 4. Experiments

We present experimental results obtained on real-life data. The first problem is to categorize text in two different category systems, and the second is multimedia categorization.

### 4.1. Text Categorization

We obtained 1213 logs describing customer problems with Xerox hardware, and the solutions given by tech support technicians. These logs were obtained from the Xerox Welcome Center (ie help desk) and are usually manually assigned two labels: one describing the *type* of problem (eg *printing*, or *network* problem, 7 categories in all) and a second for the *severity* (eg *question* or *malfunction*, 5 categories). Each incoming log must therefore be classified in two different category systems. As there is some level of dependency between the two category systems, we hope to improve the automatic classification performance using our approach.

The logs are text documents written in English. As they are often written by a rushed operator, they are usually short and contain many grammar and spelling mistakes. The only preprocessing we performed is to transform each log into a bag-of-words, based on the

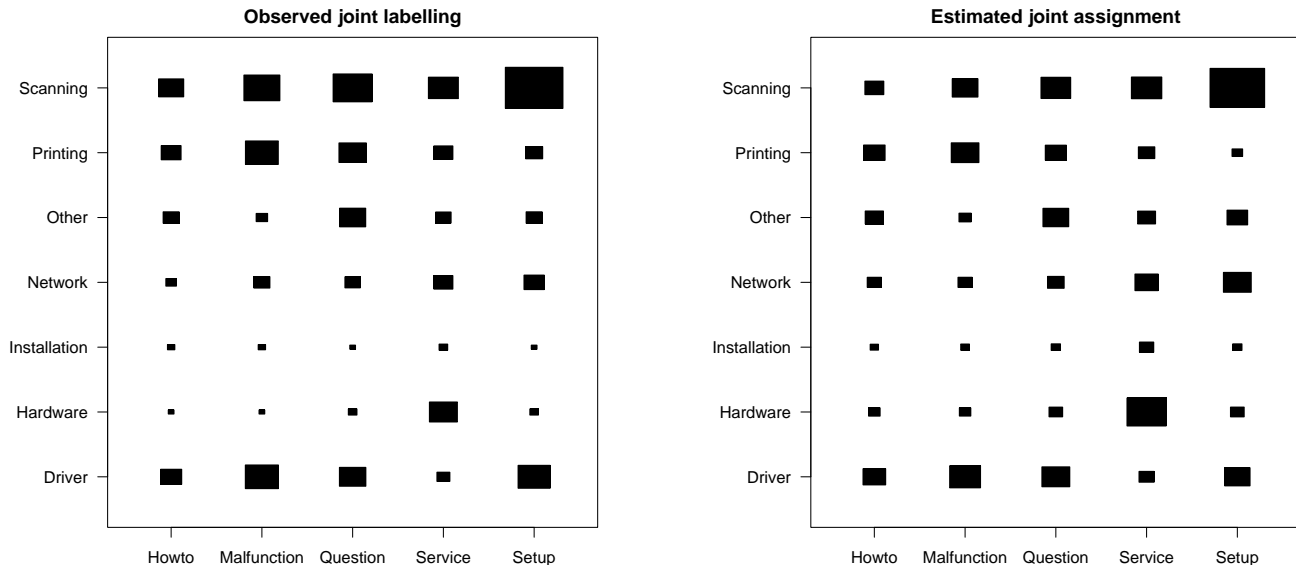


Figure 3. Observed ( $\hat{P}(c_1, c_2)$ , left) and estimated ( $\hat{P}(C^1, C^2)$ , right) joint distributions. Larger squares indicate higher probabilities.

surface forms. We removed rare words (occurring less than 3 times in the corpus) and short alphabetic words. We trained two categorizers, one on each category system. These categorizers are probabilistic latent categorizers, as described in (Gaussier et al., 2002). Exploiting these categorizers independently constitutes our baseline.

In order to estimate the sensitivity of our methods and results, we experimented with two random splits containing respectively 1016 and 1013 logs for training and 197 and 200 logs for testing, with no overlap between the test sets. The base categorizers and the reweighting factors are estimated on the training set, and the performance is computed on the test set.

Using model B, we estimate the  $7 \times 5 = 35$  reweighting factors as described in section 3.2. As an illustration, figure 3 presents  $\hat{P}(c_1, c_2)$  and  $\hat{P}(C^1, C^2)$  for the first split. Category pairs for which the observed distribution is higher than estimated, eg (Printing, Setup), will get higher factors.

The performance is reported in terms of MisClassification Error (MCE, the ratio of misclassified examples) in table 1. We test two instances of Model B. ModelB/J assigns the most probable pair of categories to the document using  $P(c_1, c_2|x)$  from eq. 5. ModelB/M marginalises the unneeded category to estimate  $P(c_1|x)$  and  $P(c_2|x)$  and assigns each category accordingly, eq. 6. As shown in table 1, Model B yields a 5 to 6% (absolute) decrease in error on the TYPE category

system, with a marginal *increase* in error on SEVERITY.

For model A, we used both the optimal solution of maximal likelihood equation 2 and its approximation  $\hat{P}(c_1, c_2)$  (on the training set) for estimating the  $\psi$  values of equation 1. It appears that the approximation gave nearly the same results as the exact optimal solution, while being less costly in computing time, so that we reported only the error rates for the approximate scheme. Moreover, this approximation seems to result in performance that is similar to the performances of Model B: Model A yields a 5% absolute decrease in error on the TYPE category system, without significant degradation on the other category system.

As a comparison, we applied the *stacking approach* using a logistic regression.<sup>4</sup> Logit/P is a single logistic regression applied to all 35 category pairs as output. Logit/R uses two logistic regression models to reweight each category system. This approach results in similar reductions in error on the TYPE category, but yields a large *increase* in error on the SEVERITY categories.

Judging from overall results, SEVERITY seems a lot more difficult to learn than TYPE. Thus, improving the latter without decreasing performance on the former is a good result.

<sup>4</sup>We also tried a Support Vector Machine, but the classification accuracy was worse than what we obtained with h logistic regression

Table 1. Performance (misclassification error, lower is better) of various multiple-view methods on two data splits, for the Type and Severity (SEV.) category systems. ModelB is used either on joint (J) or marginal (M) distributions. Logit are logistic regressions on all pairs (P) or for reweighting (R). See text for more information.

|        | MISCLASSIFICATION ERROR (IN %) |             |         |      |             |      |           |      |         |       |         |      |
|--------|--------------------------------|-------------|---------|------|-------------|------|-----------|------|---------|-------|---------|------|
|        | BASELINE                       |             | MODEL A |      | MODEL B/J   |      | MODEL B/M |      | LOGIT/P |       | LOGIT/R |      |
|        | TYPE                           | SEV.        | TYPE    | SEV. | TYPE        | SEV. | TYPE      | SEV. | TYPE    | SEV.  | TYPE    | SEV. |
| SPLIT0 | 37.6                           | 45.7        | 33.0    | 44.7 | 31.0        | 45.7 | 32.0      | 46.7 | 32.5    | 60.4  | 30.0    | 58.4 |
| SPLIT1 | 38.5                           | 47.5        | 33.5    | 49.5 | 33.0        | 48.5 | 34.5      | 49.0 | 35.5    | 53.5  | 35.0    | 51.5 |
| AVG    | 38.1                           | <b>46.6</b> | 33.2    | 47.1 | <b>32.0</b> | 47.1 | 33.2      | 47.8 | 34.0    | 57.0  | 32.5    | 54.9 |
| GAIN   |                                |             | -4.9    | +0.5 | <b>-6.1</b> | +0.5 | -4.9      | +1.2 | -4.1    | +10.4 | -5.6    | +8.3 |

## 4.2. Multimedia Categorization

For multimedia categorization, we focus on web data. The textual category system we have used has been built from Dmoz,<sup>5</sup> a.k.a. The Open Directory Project. The Open Directory Project is a directory of webpages, manually edited by a large community of volunteer editors. It is used for example by Google Directory. It offers a large spectrum of categories from which we have extracted documents related to travel and associated activities such as winter or water activities, travel preparation and transportation. We then simplified the hierarchical organization into a system of 89 categories for a maximum depth of 3 levels. On the image side, we used an internal visual category system, based on 28 generic categories which are related to the "Travel and Leisure" domain (beach, fishing, water activities, trains, planes, ...). Note that this category system includes generic objects as well as scenes. It is fundamentally multi-class, multi-label, as the same image can belong to more than one category.

The documents are web pages corresponding to web sites referenced in Dmoz. For each document we have extracted the textual information from the web page and we have collected images corresponding to the root page and the first level of the linked pages in order to have enough document with pictures. We applied a very simple filter to automatically remove irrelevant images. The filter is based on the size, the ratio between the X and Y sizes and the distribution of the colors. The goal is to remove the logos, buttons and other web decorations that populate most of the web pages.

The resulting collection contains 2828 documents, 1068 of which contain text and valid images, while the rest (1760 documents) contains only text. There are 6193 valid images in the 1068 multi-media documents.

A text categorizer was trained on these 1760 text-only documents using the Probabilistic Latent categorizer

(Gaussier et al., 2002); subsequently, this categorizer was applied to the textual part of the 1068 multimedia documents, resulting in assignment probabilities  $P(C|x)$  for each (document,category) pair.

Similarly, the 6193 images contained in the multimedia documents were scored by an image categorizer. To categorize the images we use a *Generic Visual categorization* (GVC) system based on a bag-of-keypatches approach (Csurka et al., 2004; Farquhar et al., 2005; Perronnin et al., 2006). This approach was chosen in analogy to the bag-of-words approach used for text categorization (Joachims, 1998). Similarly, an image can be characterized by a histogram of visual word counts. In contrast to the text categorization, the main difficulty for images is that we have no given visual vocabulary. Therefore, we first have to build one automatically from the training set.

In our current GVC system (Perronnin et al., 2006), we apply soft clustering based on Gaussian Mixture Models (GMM) to features consisting of orientation histogram descriptors of rotation or affine invariant regions (Mikolajczyk & Schmid, 2003), as well as color features (local mean and standard deviation computed in RGB space). Each component in the GMM becomes a visual word, and an image may be represented by a histogram over visual words. This is similar to Farquhar et al. (2005), however, in contrast to them, our class dependent vocabularies are not trained independently. Instead, we begin by building a *universal vocabulary* (GMM) from all the training data. Then, using a Bayesian adaptation of the GMM, we adapt this universal vocabulary to each class using class-specific images.

An image is then characterized by a set of bi-partite histograms - one per class - where each histogram describes whether the image content is best modeled by the universal vocabulary, or the corresponding class vocabulary. To classify these bi-partite histograms, we use one Support Vector Machine (SVM) classifier per class. The GVC system was applied independently to

<sup>5</sup><http://www.dmoz.org>

both texture features and colors features and the normalized SVM scores were merged (by taking the average) to output a score for each category, which was converted to a probability using a sigmoid fit (Platt, 1999).

As some images may have several labels, we had to extend the basic theoretical scheme described for the multi-class mono-label case, to the multi-label case. When considering the joint use of visual and textual scores to refine the visual categorizer decisions, we adopted a very simple method for this extension, by considering that the generic visual categorizer has exploited as much as possible any correlation in the visual categories and that we could limit ourselves to consider the mutual influence of the (monolabel) text categorizer with each "1-class-vs-the-rest" image categorizer. Applying the methods proposed in the theoretical sections results in refining/modifying the image categorizer scores, in order to take into account the textual context of the image. For each visual class, we then used a re-calibration procedure on a calibration set extracted from the training data, in order to ensure that the individual thresholds for each category are adapted to the new refined/modified scores. In other words, the results reported here include both effects (multiple-view correction and re-calibration).

Finally, as a document can have more than one image, when addressing the textual categorization problem using the image categorizer scores of the associated images, we furthermore have to decide how to aggregate the modifications brought by each image. We simply used the average operator to aggregate the refined scores; in other words, we applied the reweighting formula for each (document, image  $i$ ) pair, and then took the average of the refined probabilities over all images  $i$  of the same document. We then used re-calibration to obtain a more adapted threshold using the same method as for the images.

In order to assess the stability of the results, we divided the data into 5 splits in a 5-fold cross-validation manner. Note that for each fold, only the reweighting coefficients are estimated on the training part. As already mentioned, the text and image categorizers themselves are obtained beforehand on some independent data.

Table 2 presents the micro-averaged F1 measure (harmonic mean of precision and recall) we obtained on the text and image categories using various methods. Again, the baseline uses the scores of the independent categorizers without taking dependencies into account. For model B, we select the class with highest marginal (eq. 6) in each category system.

Table 2. Performance (F1 - harmonic mean of precision and recall) on the multimedia data - Average on 5 folds. ModelB is used on the marginal distributions. See text for more information.

|     | MICRO-AVERAGED F-1 MEASURE (IN %) |      |             |             |             |             |
|-----|-----------------------------------|------|-------------|-------------|-------------|-------------|
|     | BASELINE                          |      | MODEL A     |             | MODEL B     |             |
|     | TEXT                              | IMG  | TEXT        | IMG         | TEXT        | IMG         |
| AVG | 70.4                              | 51.2 | <b>72.6</b> | <b>59.3</b> | <b>71.5</b> | <b>59.7</b> |
| +/- |                                   |      | +2.2        | +8.2        | +1.1        | +8.5        |

Models A and B seem to yield uneven results for this data set: categorization performance is highly improved for the IMAGE category system (more than 8% absolute), while at the same time the F1-improvement is relatively modest for the TEXTUAL category system (about 2%). This could be explained by the fact that, for the text categorization system, the number of possible categories (89) is so high that the empirical estimate is no longer reliable for small-sized categories. The difference is statistically significant for both media, when model A or B is compared to the baseline, but not significant when models A and B are compared to each other.

## 5. Discussion

The experimental results show that our reweighting models do manage to take into account some of the dependencies between the category systems. They provide some improvement in performance, although it may not be equivalent on both category systems. In our first experiments, both models improve the classification accuracy on the TYPE categories, with limited impact on the SEVERITY categories, which seem much harder to classify accurately.

Our model may be seen as a simple instance of the general problem of calibration. Using empirical evidence from the data, we wish to calibrate the output of the independent classifiers to better match the observed dependencies. However, our investigations were limited to the classification accuracy rather than the calibration of the full output distribution.

Our approach seems especially promising for multimedia categorization problems. It allows us to leverage several existing single-media categorizers (for text, still or moving images, sound) and let the reweighting model handle the dependencies between the various category systems. This is especially valuable when one of the base category systems is also the final multimedia category system that we are interested in. One such example from our experiments would be the Dmoz category system, which may be used both for

the text categorizer and for categorizing the final multimedia content.

## 6. Conclusion

In this paper, we addressed the problem of *multiple-view categorization*, ie categorization of the same objects in multiple category systems. We furthermore focused on the practical case where we have to consider the individual categorizers (for each category system) as black-boxes; this case is the consequence of the fact that it often happens that we have numerous labeled training data when considering the category systems independently, but fewer training data annotated simultaneously in the different category systems. It is therefore hazardous to want to build a joint categorizer, based directly on the low-level features involved in a unique complex model. Resorting to simpler models, with much less parameters, is logically a more promising approach. We introduce two simple reweighting schemes (model A and model B) that rely on a limited number of parameters. We have tested these approaches in two situations: the categorization of text in two different category systems, and the categorization of multimedia content. We have showed that the best model potentially provides a large decrease in error (more than 8% absolute, 16% relative, on the image categories).

We have found that our reweighting models provide a convenient and efficient way to address the multiple-view categorization problem. These models are applicable to any combination of base classifiers, as long as their output may be converted to probabilities (using rescaling or calibration). They allow to take into account the dependencies between the different category systems, without having to retrain a large and complex classifier on the combined categories. In addition, model B and the simplified version of model A have closed-form expressions for the parameters, and therefore require no optimisation step at all.

## Acknowledgements

This work was partly supported by the IST Programme of the European Community, under the REVEAL THIS project, FP6-IST-511689, and under the PASCAL Network of Excellence, IST-2002-506778.

## References

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *COLT' 98: Proceedings of the eleventh annual conference on*

*Computational learning theory* (pp. 92–100). ACM Press.

Csurka, G., Dance, C., Fan, L., Willamowski, J., & Bray, C. (2004). Visual categorization with bags of keypoints. *Proc. ECCV International Workshop on Statistical Learning in Computer Vision*.

Farquhar, J., Szedmak, S., Meng, H., & Shawe-Taylor, J. (2005). *Improving “bag-of-keypoints” image categorisation* (Technical Report). University of Southampton.

Gaussier, E., Goutte, C., Papat, K., & Chen, F. (2002). A hierarchical model for clustering and categorising documents. *Proceedings of the 24th BCS-IRSG Colloquium on IR Research (ECIR'02)* (p. to appear). Springer.

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. *Proc. ECML* (pp. 137–142).

Mikolajczyk, K., & Schmid, C. (2003). A performance evaluation of local descriptors. *Proc. CVPR* (pp. 257–263).

Perronnin, F., Dance, C., Csurka, G., & Bressan, M. (2006). Adapted vocabularies for generic visual categorization. *Proc. ECCV*.

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. *Advances in Large Margin Classifiers*. MIT Press.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Comput. Surv.*, 34, 1–47.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5, 241–259.