Active Learning Strategies for Multi-Label Text Classification

Andrea Esuli and Fabrizio Sebastiani

Istituto di Scienza e Tecnologia dell'Informazione
Consiglio Nazionale delle Ricerche
Via Giuseppe Moruzzi 1 – 56124 Pisa, Italy
{andrea.esuli,fabrizio.sebastiani}@isti.cnr.it

Abstract. *Active learning* refers to the task of devising a ranking function that, given a classifier trained from relatively few training examples, ranks a set of additional unlabeled examples in terms of how much further information they would carry, once manually labeled, for retraining a (hopefully) better classifier. Research on active learning in text classification has so far concentrated on single-label classification; active learning for multi-label classification, instead, has either been tackled in a simulated (and, we contend, non-realistic) way, or neglected *tout court*. In this paper we aim to fill this gap by examining a number of realistic strategies for tackling active learning for multi-label classification. Each such strategy consists of a rule for combining the outputs returned by the individual binary classifiers as a result of classifying a given unlabeled document. We present the results of extensive experiments in which we test these strategies on two standard text classification datasets.

1 Introduction

In many applicative contexts involving supervised learning, labeled data may be scarce or expensive to obtain, while unlabeled data, even sampled from the same distribution, may abound. In such situations it may be useful to employ an algorithm that ranks the unlabeled examples and asks a human annotator to label a few of them, starting from the top-ranked ones, so as to provide additional training data. The task of this algorithm is thus to rank the unlabeled examples in terms of how useful they would be, once labeled, for the supervised learning task. The discipline that studies these algorithms is called *active learning* [1].

This paper focuses on the application of active learning to *text classification* (aka *text categorization* – TC), and to *multi-label* text classification (MLTC) in particular. Given a set of textual documents D and a predefined set of *classes* (aka *labels*, or *categories*) $C = \{c_1, \ldots, c_m\}$, MLTC is defined as the task of estimating an unknown *target function* $\Phi : D \times C \rightarrow \{-1, +1\}$, that describes how documents ought to be classified, by means of a function $\hat{\Phi} : D \times C \rightarrow \{-1, +1\}$ called the *classifier*[1]; here, $+1$ and -1 represent membership and non-membership of the document in the class. Each document may thus belong to

[1] Consistently with most mathematical literature we use the caret symbol (ˆ) to indicate estimation.

M. Boughanem et al. (Eds.): ECIR 2009, LNCS 5478, pp. 102–113, 2009.

zero, one, or several classes at the same time. MLTC is usually accomplished by generating m independent binary classifiers $\hat{\Phi}_j$, one for each $c_j \in C$, each entrusted with deciding whether a document belongs or not to class c_j.

In this paper we will restrict our attention to classifiers that, aside from taking a binary decision on a given document, also return as output a *confidence estimate*, i.e., a numerical value representing the strength of their belief that the returned decision is correct. We formalize this by taking a classifier to be a function $\hat{\Phi} : D \times C \rightarrow [-1, +1]$ in which the sign of the returned value $sgn(\hat{\Phi}(d_i, c_j))$ indicates the decision of the classifier, and the absolute value $|\hat{\Phi}(d_i, c_j)|$ represents its confidence in the decision (the higher the value, the higher the confidence).

MLTC is different from *single-label* TC (SLTC) since this latter tackles the case in which one and only one class must be attributed to each document. This is formalized by viewing a classifier as a function $\hat{\Phi} : D \rightarrow C \times [0, 1]$ which, given a document, returns the class to which the classifier believes the document to belong, plus an estimate of the classifier's confidence in this belief.

An analysis of previous work on active learning in TC (see Section 4) shows that this literature has so far exclusively concentrated on SLTC. In this context, a typical strategy for active learning consists, once a classifier has been generated with the available training examples, of ranking the unlabeled examples in increasing order of the confidence that this classifier had in classifying them, since an example which the system classified with low confidence has a high probability of being, once labeled by a human annotator, very informative for retraining the classifier (see e.g. [2]).

However, it is of key importance to note that this strategy is only made possible by the fact that in SLTC a single confidence value is returned for each unlabeled example. Conversely, in MLTC this strategy cannot be applied straightforwardly, since for each test document d_i MLTC generates m different confidence values $|\hat{\Phi}(d_i, c_j)|$, one for each $c_j \in C$. This means that either

1. m independent document rankings are generated, each based on the confidence scores returned by a given binary classifier $\hat{\Phi}_j$, after which the human annotator scans each class-specific ranking, one by one, annotating for each such ranking the top-ranked documents. We call this option *local labeling*, since the labeling activity is performed locally to each class. Or:
2. a unique ranking is generated, based on the combination of the m confidence scores associated to the same document. We call this option *global labeling*, since the labeling activity is performed globally to the entire set of classes.

Local labeling has been frequently adopted, in a simulated way, in laboratory research on active learning. However, we argue that this is not feasible in practice. In fact, let us assume that the average human effort involved in reading (or browsing through, or understanding for the sole purpose of classifying) a document is r, and that the average human effort involved in deciding whether a given class should be attributed or not to this document is $c \ll r$ (we here assume that an annotator already has an understanding of the meaning of the classes); then the total effort involved in classifying a document is $r + m * c$. The key observation here is that, in all likelihood, $(r + m * c) < 2(r + c) \ll m(r + c)$ for

any reasonable value of m; that is, deciding which among the m classes should be attributed to a document we have read requires less effort than reading it again, and much less effort than reading it m times!

Local labeling is infeasible exactly because it would require a human annotator to scan m different rankings, and hence to examine the same unlabeled document up to m times in order to label it. Note that m may be large or very large: it may be in the hundreds (as, e.g., in the REUTERS-21578 [3] and RCV1-v2 [4]), but it may also be in the hundreds of thousands (as in the YAHOO! collection [5]). In operational environments one is thus left with only global labeling as an option; it is different combination strategies for global labeling that this paper proposes and studies experimentally.

We remark that this paper does *not* deal with active learning algorithms for specific supervised learning devices (such as e.g., [6]), but presents active learning strategies that are independent of the learning device, and that are suitable for use with any such device. Incidentally, we note that this is the first work that performs a truly large-scale experimentation of active learning in TC, since previous works [2,6,7,8,9,10] have only addressed small datasets, with few test documents, or few classes, or both. To the contrary, we here investigate active learning in the context of two standard MLTC collections, both including approximately 100 classes, one of them including almost 800,000 test documents.

The rest of the paper is organized as follows. Our strategies for performing active learning in MLTC are described in Section 2. Section 3 discusses our experiments and the experimental protocol we have followed. We review related work in Section 4 and conclude in Section 5 by discussing future work.

2 Active Learning Strategies for MLTC

In this work we compare several strategies for ranking the automatically labeled documents and presenting them to a human annotator for global labeling. We explore three orthogonal dimensions according to which a given strategy σ may be designed; we call them the "evidence" dimension, the "class" dimension, and the "weight" dimension. Each individual strategy will thus result from making a choice among several possible alternatives for each of the three dimensions.

From now on, as a notational convention, a given ranking strategy σ is identified by a sequence of three capital boldface letters, each letter indicating a choice made according to a given dimension. For instance, the sequence **SAN** will denote a strategy obtained by choosing MAXSCORE (**S**) for the "evidence" dimension, AVG (**A**) for the "class" dimension, and NOWEIGHTING (**N**) for the "weight" dimension (see Sections 2.1 to 2.3 for the precise meaning of these choices); 2 choices are available for the "evidence" dimension, 3 for the "class" dimension, and 2 for the "weight" dimension, giving rise to $2 * 3 * 2 = 12$ different strategies. We will also use the "$*$" symbol as a wildcard, so that, e.g., the sequence **SA*** will denote the *set* of the two strategies obtained by choosing MAXSCORE (**S**) for the "evidence" dimension, AVG (**A**) for the "class" dimension, and either of the two available choices for the "weight" dimension.

We will also use the following terminology. Given a classifier $\hat{\Phi} : D \times C \rightarrow [-1, +1]$, the value $\hat{\Phi}(d_i, c_j)$ will be called the *c_j-score of d_i*; the value $|\hat{\Phi}(d_i, c_j)|$ will be called the *c_j-confidence of d_i*; and the value $sgn(\hat{\Phi}(d_i, c_j))$ will be called the *c_j-sign of d_i*. We will further assume that we have a policy for combining these class-dependent values into a single class-independent value (how this policy may vary is exactly the topic of Section 2.2); accordingly, the value $\hat{\Phi}(d_i)$ will be called the *score of d_i*; the value $|\hat{\Phi}(d_i)|$ will be called the *confidence of d_i*; and the value $sgn(\hat{\Phi}(d_i))$ will be called the *sign of d_i*.

We now move to discussing the three above-mentioned dimensions in detail.

2.1 The "Evidence" Dimension

The "evidence" dimension has to do with the type of evidence we decide to use as a basis for ranking the unlabeled documents.

One potential choice is to use as evidence the confidence value $|\hat{\Phi}(d_i)|$ with which the unlabeled document d_i has been classified. As mentioned in Section 1, the underlying intuition is that the lower the confidence value, the more the document should prove informative for retraining the classifier, which means that the documents which minimize this confidence value should be the top-ranked ones. As a consequence, we call this choice MinConfidence (in symbols: **C**); essentially, this corresponds to the notion of *uncertainty sampling* discussed in [2] (see Section 4). Of course, the catch here is that, in reality, not a single confidence value $|\hat{\Phi}(d_i)|$, but m different c_j-confidence values $|\hat{\Phi}(d_i, c_j)|$, are generated for each unlabeled document d_i. Exactly how these c_j-confidence values should generate "the" confidence value of d_i according to which the ranking should be produced is the topic of the "class" dimension, to be discussed in Section 2.2.

A second, alternative choice is instead to use as evidence the score $\hat{\Phi}(d_i)$ returned for d_i by the classifier. Here a different intuition is at play, namely, that the higher the score, the more likely it is that d_i is a positive example (since scores close to 1 indicate high confidence that the document is a positive example, and scores close to -1 indicate high confidence that the document is a negative one), and that it is exactly positive examples, rather than negative ones, that are typically most useful in a supervised learning task. As a consequence, we call this choice MaxScore (**S**); essentially, this corresponds to the notion of *relevance sampling* discussed in [2] (see Section 4). Again, we are faced with the fact that m different c_j-scores are generated for each unlabeled document d_i; again, exactly how these c_j-scores should generate "the" score of d_i according to which the ranking should be produced, will be discussed in Section 2.2.

2.2 The "Class" Dimension

The "class" dimension has to do with the fact that, whatever type of evidence we elect to use (as from the "evidence" dimension), for each automatically labeled document d_i there are m different values for this evidence, one for each class $c_j \in C$; each alternative choice for this dimension represents a policy on how to generate one class-independent piece of evidence from the m class-specific ones.

One potential choice is picking the value that maximizes our expected informativeness across all $c_j \in C$. If our choice according to the "evidence" dimension is MinConfidence, this will mean picking $\min_{c_j \in C} |\hat{\Phi}(d_i, c_j)|$, i.e., the minimum across the c_j-confidence values; if we have instead gone the MaxScore route, then this will mean picking $\max_{c_j \in C} \hat{\Phi}(d_i, c_j)$, i.e., the maximum among the c_j-scores. The rationale of this policy is that we want the manual annotator to concentrate on the documents that are deemed to be extremely valuable at least for one class. We call this choice Min/Max (**M**).

A second, alternative choice is averaging all values across all $c_j \in C$. This policy is intended to force the human annotator to label the documents deemed to be at least fairly valuable for many classes. We call this choice Avg (**A**).

A further, alternative choice consists in employing a *round robin* policy, according to which the top-ranked examples for each class are picked, so that each class will be adequately championed in the resulting rank. This is obtained by (a) picking, for each class $c_j \in C$, the best automatically labeled document according to the criterion chosen for the "evidence" dimension, (b) ranking these m documents according to this criterion, (c) using the resulting ranking to fill the positions from the 1st to at most the m-th of the global rank. After this, these three steps are repeated a second time by ranking the second best documents for each class and using the resulting ranking to fill the positions from at most the $m + 1$-th to at most the $2m$-th of the global rank; ... after which the three steps are repeated a k-th time by ranking the k-th best documents for each class and using the resulting ranking to fill the positions from at most the $((k-1)m + 1)$-th to at most the km-th of the global rank[2]. We call this choice RoundRobin (**R**).

2.3 The "Weight" Dimension

The "weight" dimension has to do with the fact that, in ranking the unlabeled documents, it might or it might not be desirable to treat all classes equally.

One choice is to give more weight to those classes on which the current classifier is still performing badly, so as to prefer those documents that are likely to bring about an improvement where it is most needed. Assume we are using an evaluation function $f(\hat{\Phi}_j)$ that ranges on $[0, 1]$ (with higher values indicating better effectiveness). This policy thus corresponds (i) to multiplying the $|\hat{\Phi}(d_i, c_j)|$ confidence value by $f(\hat{\Phi}_j)$ (which indicates the effectiveness that the current classifier has obtained on class c_j) in case MinConfidence is the choice for the "evidence" dimension, or (ii) to multiplying the $\hat{\Phi}(d_i, c_j)$ score by $(1 - f(\hat{\Phi}_j))$ in case MaxScore has been chosen instead. Note that when $f(\hat{\Phi}_j) = 0$ (resp., $f(\hat{\Phi}_j) = 1$), for the MinConfidence strategy (resp., for the MaxScore strategy) the multiplier defined by the weight dimension would be equal to 0; if Min/Max were the choice for the "class" dimension, this would result in all

[2] Duplicates are obviously removed. That is, when the same document is selected for different classes, in the same round on in different rounds, it is used only once in the global ranking; in this case, strictly less than km documents will be ranked.

documents having the same rank, which is undesirable. We have solved this issue by always using, for the purposes of the weight dimension, Laplace-smoothed estimates of F_1, with the smoothing parameter set to $\epsilon = 0.05$.

Since our evaluation measure of choice will be F_1, we call this choice F_1-WEIGHTING (**W**). An alternative choice is instead to treat all classes alike. We call this choice NOWEIGHTING (**N**).

3 Experiments

As the learning device for generating our classifiers we have used a boosting-based learner, called MP-BOOST [11]; boosting is currently among the classes of supervised learning devices that obtain the best performance in a variety of learning tasks and, at the same time, have strong justifications from computational learning theory. MP-BOOST is a variant of ADABOOST.MH [12] optimized for multi-label settings, which has been shown [11] to obtain considerable effectiveness improvements with respect to ADABOOST.MH. In all the experiments the algorithm has been run with a number of iterations fixed to 1,000.

As datasets, in our experiments we have used the REUTERS-21578 and RCV1-V2 corpora. REUTERS-21578 is probably still the most widely used benchmark in MLTC research[3]. It consists of a set of 12,902 news stories, partitioned (according to the "ModApté" split we have adopted) into a training set of 9,603 documents and a test set of 3,299 documents. The documents are labelled by 118 categories; in our experiments we have restricted our attention to the 115 categories with at least one positive training example. REUTERS CORPUS VOLUME 1 version 2 (RCV1-V2)[4] is a more recent MLTC benchmark made available by Reuters and consisting of 804,414 news stories produced by Reuters from 20 Aug 1996 to 19 Aug 1997. In our experiments we have used the "LYRL2004" split, defined in [4], in which the (chronologically) first 23,149 documents are used for training and the other 781,265 are used for test. Of the 103 "Topic" categories, in our experiments we have restricted our attention to the 101 categories with at least one positive training example. Consistently with the evaluation presented in [4], also categories placed at internal nodes in the hierarchy are considered in the evaluation; again, consistently with [4], as positive training examples of these categories we use the union of the positive examples of their subordinate nodes, plus their "own" positive examples.

In all the experiments discussed in this paper stop words have been removed, punctuation has been removed, all letters have been converted to lowercase, numbers have been removed, and stemming has been performed by means of Porter's stemmer. Word stems are thus our indexing units; since MP-BOOST requires binary input, only their presence/absence in the document is recorded, and no weighting is performed.

As a measure of effectiveness that combines the contributions of *precision* (π) and *recall* (ρ) we have used the well-known F_1 function, defined as $F_1 = \frac{2\pi\rho}{\pi+\rho} =$

[3] `http://www.daviddlewis.com/resources/testcollections/~reuters21578/`
[4] `http://trec.nist.gov/data/reuters/reuters.html`

$\frac{2TP}{2TP+FP+FN}$, where TP, FP, and FN stand for the numbers of true positives, false positives, and false negatives, respectively. Note that F_1 is undefined when $TP = FP = FN = 0$; in this case we take F_1 to equal 1, since the classifier has correctly classified all documents as negative examples. We compute both microaveraged F_1 (denoted by F_1^μ) and macroaveraged F_1 (F_1^M). F_1^μ is obtained by (i) computing the category-specific values TP_i, FP_i and FN_i, (ii) obtaining TP as the sum of the TP_i's (same for FP and FN), and then (iii) applying the $F_1 = \frac{2TP}{2TP+FP+FN}$ formula. F_1^M is obtained by first computing the category-specific F_1 values and then averaging them across the c_j's.

3.1 Experimental Protocol

In this work we adopt the following iterative experimental protocol; the protocol has three integer parameters α, β, and γ. Let Ω be a dataset partitioned into a training set Tr and a test set Te, and let σ be an active learning strategy:

1. Set an iteration counter $t = 0$;
2. Set the current training set Tr_t to the set of the chronologically[5] first α examples of Tr; set the current "unlabeled set" $U_t \leftarrow Tr/Tr_t$;
3. For $t = 1, \ldots, \beta$ repeat the following steps:
   (a) Generate a classifier $\hat{\Phi}^t$ from the current training set Tr_t;
   (b) (If σ is one of the strategies in ****W**) Evaluate (by means of F_1) $\hat{\Phi}^t$ by 5-fold cross-validation on Tr_t;
   (c) Evaluate the effectiveness of $\hat{\Phi}^t$ on Te;
   (d) Classify U_t by means of $\hat{\Phi}^t$;
   (e) Rank U_t according to strategy σ (if σ is one of the strategies in ****W**, the F_1 values required by the strategy are those computed at Step 3b);
   (f) Let $r(U_t, \gamma)$ be the set of the γ top-ranked elements of U_t; set $Tr_{t+1} \leftarrow Tr_t \cup r(U_t, \gamma)$; set $U_{t+1} \leftarrow U_t/r(U_t, \gamma)$.

We remark that Step 3c has only the purpose of collecting the results for experimental purposes (i.e., for producing the tables of Section 3.2); since it uses the test set Te, its results are obviously in no way accessible to the algorithm.

The above protocol simulates the work of a human annotator who has available a training set Tr_0 consisting of α training examples, and an "unlabeled set" U_0 consisting of $|Tr| - \alpha$ unlabeled examples. The annotator generates a classifier $\hat{\Phi}^0$ from Tr_0, uses it to classify the documents in U_0, asks the active learning agent to rank them, manually labels the γ top-ranked ones, generates a new classifier $\hat{\Phi}^1$ from an augmented training set that comprises Tr_0 and the γ newly labeled examples, and repeats this process β times.

In all our experiments we have set, for both datasets, $\alpha = 100$, $\beta = 20$, and $\gamma = 50$; this means that each strategy will be evaluated by testing the accuracy of the classifiers generated from training sets consisting of 100, 150, ..., 950, 1000 training examples, for a total 19 experiments per strategy. We think these

[5] Our two datasets consist of news stories that were broadcast by Reuters over a period of time; "chronological order" here refers to the date of issue of these news stories.

parameters are realistic, since they simulate a situation in which: (i) there are only 100 training examples at the beginning; (this is reasonable, since in many applications in which significantly more training examples are available, human annotators might not find it worthwhile to annotate any further); (ii) every time the human annotator manually labels 50 unlabeled examples, he/she wants to retrain the system (this is reasonable, since (a) he/she wants to check whether the added training examples have increased the accuracy of the system (this can be done by having the system always perform Step 3b), and since (b) he/she wants the operate on a ranking of the unlabeled documents that incorporates as much as possible the feedback he/she has already given to the system); (iii) the human annotator does not want to do any further manual labeling once 1,000 training examples are available (this seems reasonable, since at this point the cost-effectiveness of the manual effort has probably decreased significantly).

As the baseline strategy for the evaluation of our results we adopt the one that consists in adding further labeled documents to the training set by picking them at random. This simulates the behaviour of a human annotator that picks unlabeled documents and labels them in no particular order.

3.2 Results and Discussion

The main results of our experiments are summarized in Table 1. The top 4 rows report, for each individual strategy, the values of F_1^{μ} and F_1^{M} obtained by averaging across the results of the 19 different training sessions resulting from running the protocol of Section 3.1 with $\alpha = 100$, $\beta = 20$ e $\gamma = 50$. The bottom 4 rows focus instead on the last among these 19 values, i.e., reports the F_1^{μ} and F_1^{M} values obtained by the various classifiers trained on the 1,000 training examples available by the end of the active learning process. Table 2 is obtained by averaging the values from Table 1 (top 4 rows) across all possible values for two of the three dimensions of Sections 2.1 to 2.3, so as to allow a direct comparison among the various possible choices for the same dimension. In order to validate the relevance of the results produced by our strategies with respect to the baseline, we have subjected to a statistical significance macro t-test [13] the results produced by the final classifiers trained on 1,000 examples (i.e., those reported in the bottom 4 rows of Table 1); all the results have turned out to be statistically significantly different from the baseline at a p-value ≤ 0.01.

It is clear from these tables that the results are not easy to interpret. Table 1 (top 4 rows) shows that no single strategy clearly emerges as the winner. For Reuters-21578, **CMW** emerges as the best in terms of F_1^{μ}, but the best in terms of F_1^{M} is a completely different strategy, namely, **SAN**; for RCV1-v2, instead, yet a third strategy proves the best (namely, **CMN**), this time for both F_1^{μ} and F_1^{M}.

The situation becomes a bit clearer by looking at Table 2, which allows us to appreciate the contribution of the various dimensions to the overall process.

The first indication we receive from Table 2 is that, in terms of the "evidence" dimension, using the confidence of d_i (MinConfidence) is more useful than using its score (**S**), since **C**** strategies outperform **S**** strategies for both

Table 1. Values of F_1 averaged across the 19 different training sessions (top 4 rows), and values of F_1 obtained in the last training session, i.e., with 1,000 training examples selected as a result of the active learning strategy (bottom 4 rows). **Boldface** indicates the best performance on the dataset.

		Base	**CMW**	**SMW**	**CAW**	**SAW**	**CRW**	**SRW**	**CMN**	**SMN**	**CAN**	**SAN**	**CRN**	**SRN**
F_1^μ	REUTERS-21578	.682	**.722**	.631	.683	.657	.698	.687	.704	.671	.673	.692	.708	.689
	RCV1-v2	.530	.511	.470	.491	.485	.506	.471	**.566**	.514	.513	.493	.541	.493
F_1^M	REUTERS-21578	.541	.542	.508	.552	.531	.522	.534	.543	.535	.558	**.559**	.564	.549
	RCV1-v2	.236	.215	.166	.198	.186	.215	.186	**.261**	.224	.224	.188	.229	.176
F_1^μ	REUTERS-21578	.752	**.790**	.696	.771	.755	.777	.752	.765	.748	.747	.783	.769	.750
	RCV1-v2	.622	.599	.503	.598	.565	.583	.522	**.639**	.570	.594	.575	.624	.560
F_1^M	REUTERS-21578	.575	.595	.547	.615	.600	.578	.576	.570	.597	.617	**.642**	.617	.607
	RCV1-v2	.304	.272	.183	.284	.247	.270	.230	**.312**	.274	.276	.261	.299	.224

Table 2. Values of F_1 averaged across the 19 different training sessions and across two of the three dimensions. **Boldface** indicates the best performance on the dataset across the same dimension.

			evidence		class			weight	
		Base	**C****	**S****	***M***	***A***	***R***	****W**	****N**
F_1^μ	REUTERS-21578	.682	**.698**	.671	.682	.676	**.695**	.680	**.689**
	RCV1-v2	.530	**.521**	.488	**.515**	.495	.503	.489	**.520**
F_1^M	REUTERS-21578	.541	**.547**	.536	.532	**.550**	.542	.532	**.551**
	RCV1-v2	.236	**.224**	.188	**.216**	.199	.202	.194	**.217**

datasets and both measures. This means that the principle according to which we should encourage the labeling of documents on which the current classifier is very uncertain, is *more* powerful than the principle according to which we should maximize the influx of new positive examples. This is not surprising. In fact, the intuition that underlies the former principle is that documents on which the current classifiers are very uncertain lie near the surface that, in feature space, separates positive from negative examples according to the current classifiers, and that, as a consequence, knowing on which side of the surface these documents *actually* lie allows the learning device to individuate a better-fitting surface. Conversely, while adopting the latter principle indeed tends to maximize the influx of new positive examples, these positive examples tend to be rather uninformative, since the current classifiers were already fairly convinced of their positivity; thus, having them labeled by the human annotator tends to reinforce the classifiers in their already held beliefs, but does not improve much the insight of the classifiers on *different* types of examples. From an experimental point of view, a similar conclusion had been reached already in [2] (see Section 4); our experiments thus confirm the results of [2] on a much larger experimental scale.

A second indication we receive from Table 2 is that, in terms of the "weight" dimension, treating all classes alike (NOWEIGHTING) is better than weighting them according to how bad the current performance of the corresponding classifier is (F_1-WEIGHTING). This is somehow more surprising, but can probably be explained by the fact that the F_1^μ and F_1^M measures indeed treat all classes

alike[6]; therefore, a policy, such as NoWeighting, that treats all classes alike may be seen as directly optimizing the chosen effectiveness measures.

Indications are less clear concerning the "value" dimension; Min/Max is the best performing policy on RCV1-v2, both for F_1^{μ} and for F_1^{M}, while on Reuters-21578 the winners are Avg for F_1^{M} and RoundRobin for F_1^{μ}. While none among these three policies emerges as the clear winner, we believe Min/Max should be the policy of choice, since it is the best performer, *and for both measures*, on the larger of the two test collections; proving the best on the 780,000+ test documents of RCV1-v2 should indeed be considered stronger evidence than proving the best on the 3,000+ test documents of Reuters-21578.

4 Related Work

Several works have addressed active learning in the context of text classification applications. Lewis and Gale [2] propose *uncertainty sampling* (US), which consists in ranking unlabeled documents in increasing order of their c_j-confidence. The authors compare US with *relevance sampling* (RS), i.e., ranking unlabeled documents in decreasing order of their c_j-score, and find that US outperforms RS. Liere and Tadepalli [8] test various *query by committee* strategies, whereby a committee of classifiers classify the unlabeled examples, and those on which the members of the committee disagree most are ranked highest. McCallum and Nigam [9] further combine Liere and Tadepalli's query-by-committee method with *Expectation Maximization* (EM) in order to take full advantage of the word co-occurrence information that can be mined from the unlabeled documents. Tong and Koller [6] propose an active learning method specific to SVMs, in which ranking unlabeled documents is based on *version space minimization* through various margin selection criteria. Xu et al. [10]'s *representative sampling* method is based on clustering the unlabeled documents that lie inside the margin determined by the SVM model learned in the previous iteration. After m clusters are identified, the m "medoid" documents are added to the training set. Hoi et al. [14] explore the problem of selecting an optimal *batch* of k unlabeled documents at each iteration, so as to avoid the possibility that the set of the k unlabeled documents top-ranked by an active learning process contain redundant information, as when this set contains near-duplicates. For this they propose to select the set of k documents that minimizes the global amount of redundancy, as measured by the Fisher information of the classification model. Davy and Luz [7] propose two "history-based" selection strategies. Their *history uncertainty sampling* (HUS) strategy is an extension of Lewis and Gale's [2] US strategy in which the ranking value for a document is the sum of US values obtained in the last k iterations of the active learning process. Their *history*

[6] It might be argued that F_1^{μ} does *not* treat all classes alike, since more frequent classes weight more. However, it is not class frequency that F_1-Weighting pays attention to, but effectiveness of the current classifier on the class. It is thus possible that, had we devised an alternative choice to NoWeighting and F_1-Weighting that emphasized more frequent classes, this might have excelled in terms of F_1^{μ}.

Kullback-Leibler divergence (HKLD) is instead a strategy that tends to select the documents that have been labeled erratically by the most recently generated classifiers. Finally, the work of Raghavan et al. [15,16] focuses on active learning as the task of simultaneously ranking *features and documents* for human annotation, for the purpose of improving feature selection.

One common feature of all the works discussed above is that, when they test their method on a multi-label collection with m classes, they run m independent binary experiments, thus simulating a local labeling method (which, we have argued, is artificial and unrealistic). A second common feature of all these works is that the scale of the experiments they carry out is much smaller than in the present paper, since they all test their methods on no more than 20,000 documents ([2] is the exception, with a test set of about 50,000 documents), and on no more than 10 classes. On the contrary, we work on more than 100 classes for each dataset, and use one dataset with more than 780,000 test documents; the present paper thus qualifies as the first truly large-scale experimentation on active learning in text classification.

We should also remark that, to our knowledge, active learning for multi-label classification has never been addressed even outside the realm of *text* classification; the reason of this is the fact that the machine learning literature is usually concerned with single-label classification, and tends to consider multi-label classification as a trivial reiteration of binary (hence single-label) classification.

5 Conclusions

Previous works in active learning in multi-label text classification have made the assumption that the unlabeled examples are ranked and presented to the human annotator m times, one per class. We have argued that this is unrealistic, since m is often in the hundreds at the very least, and this "local labeling" approach would likely require the human annotator to examine the very same unlabeled document more than once, in the context of different rankings. As a consequence, we have examined a set of more realistic strategies for "global labeling", i.e., for generating a single ranking of the unlabeled documents that combines the m different sources of evidence, one per class, available for the same document. We have studied 12 such strategies in a large-scale experimental study, and argued for the superiority of one such strategy, **CMN**.

In the near future we plan to extend this work by studying how this best-performing strategy behaves as a function of the parameters α, β and γ of Section 3.1, and as a function of the relationship of these parameters with the number m of classes in the dataset.

References

1. Cohn, D., Atlas, L., Ladner, R.: Improving generalization with active learning. Machine Learning 15(2), 201–221 (1994)
2. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval (SIGIR 1994), Dublin, IE, pp. 3–12 (1994)

3. Lewis, D.D.: Reuters-21578 text categorization test collection Distribution 1.0 README file, v 1.3 (2004)
4. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: RCV1: A new benchmark collection for text categorization research. Journal of Machine Learning Research 5, 361–397 (2004)
5. Liu, T., Yang, Y., Wan, H., Zeng, H., Chen, Z., Ma, W.: Support vector machines classification with a very large-scale taxonomy. SIGKDD Explorations 7(1), 36–43 (2005)
6. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. Journal of Machine Learning Research 2, 45–66 (2001)
7. Davy, M., Luz, S.: Active learning with history-based query selection for text categorisation. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECiR 2007. LNCS, vol. 4425, pp. 695–698. Springer, Heidelberg (2007)
8. Liere, R., Tadepalli, P.: Active learning with committees for text categorization. In: Proceedings of the 14th Conference of the American Association for Artificial Intelligence (AAAI 1997), Providence, US, pp. 591–596 (1997)
9. McCallum, A.K., Nigam, K.: Employing EM in pool-based active learning for text classification. In: Proceedings of the 15th International Conference on Machine Learning (ICML1998), Madison, US, pp. 350–358 (1998)
10. Xu, Z., Yu, K., Tresp, V., Xu, X., Wang, J.: Representative sampling for text classification using support vector machines. In: Sebastiani, F. (ed.) ECIR 2003. LNCS, vol. 2633, pp. 393–407. Springer, Heidelberg (2003)
11. Esuli, A., Fagni, T., Sebastiani, F.: MP-boost: A multiple-pivot boosting algorithm and its application to text categorization. In: Crestani, F., Ferragina, P., Sanderson, M. (eds.) SPIRE 2006. LNCS, vol. 4209, pp. 1–12. Springer, Heidelberg (2006)
12. Schapire, R.E., Singer, Y.: Boostexter: A boosting-based system for text categorization. Machine Learning 39(2/3), 135–168 (2000)
13. Yang, Y., Liu, X.: A re-examination of text categorization methods. In: Proceedings of the 22nd ACM International Conference on Research and Development in Information Retrieval (SIGIR 1999), Berkeley, US, pp. 42–49 (1999)
14. Hoi, S.C.H., Jin, R., Lyu, M.R.: Large-scale text categorization by batch mode active learning. In: Proceedings of the 15th International Conference on World Wide Web (WWW 2006), Edinburgh, UK, pp. 633–642 (2006)
15. Raghavan, H., Madani, O., Jones, R.: InterActive feature selection. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005), Edinburgh, UK, pp. 841–846 (2005)
16. Raghavan, H., Madani, O., Jones, R.: Active learning with feedback on features and instances. Journal of Machine Learning Research 7, 1655–1686 (2006)