Design and Analysis of the Nomao challenge Active Learning in the Real-World

Laurent Candillier[1] and Vincent Lemaire[2]

[1] Nomao - Ebuzzing Group, 1 av. Jean Rieux, 31500 Toulouse
[2] Orange Labs, 2 av. Pierre Marzin, 22300 Lannion

Abstract. *Active Learning* is an active area of research in the *Machine Learning* and *Data Mining* communities. In parallel, needs for efficient active learning methods are raised in *real-world* applications. As an illustration, we present in this paper an active learning challenge applied to a real-world application named *Nomao*. *Nomao* is a search engine of places. It aggregates information coming from multiple sources on the web to propose complete information related to a place. In this context, active learning is used to efficiently detect data that refer to a same place. The process is called *data deduplication*. Since it is a real-world application, some additional constraints have to be handled. The main ones are scalability of the proposed method, representativeness of the training dataset, and practicability of the labeling process.

1 Brief introduction to active learning

Active learning methods come from a parallel between active educational methods and learning theory [1]. The learner is from now a statistical model instead of a student. The interactions between the student and the teacher correspond to the opportunity for the model to interact with a human expert. The examples are situations used by the model to generate knowledge on the problem. Active learning methods allow the model to interact with its environment by selecting the more "informative" situations.

This paper restricts the active learning domain to the machine learning paradigm[3]. The purpose is to train a model which uses as few examples as possible. The elaboration of the training set is done in interaction with a human expert to maximize the progress of the model. The model must be able to detect the more informative examples for its learning and to ask to the expert: "what should be done in these situations ?".

Two scenarios are possible if one considers the raw data or the data descriptors. These two scenarios are adaptive sampling [3] and selective sampling [4]. The Nomao problem described in the next section commands to use the selective sampling where the model observes only a restricted part of the universe materialized by training examples stripped of label. Consequently, the input vectors selected by the model always correspond to a raw data. The image of a "bag" of instances for which the model can ask labels is usually used.

[3] The reader may find a more comprehensible survey on active learning in [2].

2 Nomao - problem description

Nomao[4] is a search engine of places that ranks results according to what you like and what your social network members like. Its development raises many scientific issues: extraction and structuralization of local content, query understanding and information retrieval, results ranking, personalization and recommendations [5].

During its first step of content extraction, Nomao collects data coming from multiple sources from the web and needs to aggregate them properly. First task consists in detecting what data refer to the same place. To automate this *deduplication* process and avoid hand-coded functions to resolve the various data inconsistencies, a *Machine Learning* method is used. Being given a set of pairs of records labeled as referring to the same place or not, a predictive model is built that is then able to decide if two records should be merged or not.

One key challenge is then to find a relevant set of training examples to be provided to the classifier. We reach here the domain of *Active Learning* [6]. Since it is conducted on real data, some specific issues are raised. The main ones are scalability of the proposed active learning method, representativeness of the training dataset, and practicability of the labeling process.

2.1 Real-world issues

Nomao dataset contains millions of examples. First of all, the proposed method must be able to handle such high volumes of data. It must also be able to perform a learning phase on training data using only a small part of the entire distribution. Indeed, the sampling of the dataset may not be uniform since it could perhaps not cover the entire domain space.

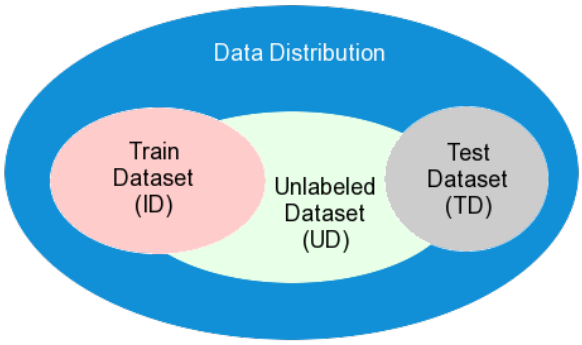

Fig. 1. Learning when test and train inputs can have different distributions

This is illustrated in Figure 1. An initial training dataset (ID) of 29,104 examples, illustrated by the red circle (on the left), had been built *by hand* by an expert of Nomao,

[4] `http://www.nomao.com/`

following his *feelings*. Then 1,985 examples have been drawn randomly and labeled in order to create a test dataset (TD). Besides, 100,000 examples have been randomly selected to create an Unlabeled Dataset (UD). The entire distribution (containing all data) could thus be the blue circle (global), UD the green one (middle), and TD the grey one (right), that could be disjointed from the red one of ID. So the results obtained on the initial training dataset could differ from the one obtained on the test dataset.

There are a lot of methods or statistical tests to examine if test, train and unlabeled inputs have different distributions. One simple way is to train a classifier where the train inputs belong to a (imaginary) positive class and the test examples to a (imaginary) negative class. If the classifier is robust and able to separate the two distributions then the degree of performances of this classifier is a good indication. In the real-world problem presented in this paper, the distributions are somewhat different: see Section 2.3 below.

The other important issue carried by this real-world application is the *practicability* of the labeling process. Indeed, with such high volumes of data, following the classical way of running active learning (labeling examples one by one and updating the model at each step) is unpractical. It is too long and too time-consuming for the labeling expert. So sets of examples must be proposed for labeling rather than individual examples. That is known as the problem of *purchasing data labels by batches*, and it has been shown in [7] that in that case, the number of examples labeled at each iteration of a procedure of active learning influences the quality of the involved model.

At last, one feature of interest of the Nomao challenge is that it involves a real human labeling of data from samples selected by the competitors, contrary to other challenges where the labeling phases have been simulated. We will see in the analysis of the results on Section 4.3 that this will help us understand what is behind the different active phases performed.

2.2 Data format

Available Nomao raw material is spots (places) descriptions. A spot is defined by the following main features: name, address, geolocalization (GPS), website, phone, fax, etc. (but data may be wrong or missing). Consider for instance the (partial) spot raw material provided in table 1.

ID	Name	Phone	Address	GPS
1	La poste	3631	13 Rue De La Clef 59000 Lille France	(50.64, 3.04)
2	La poste	0320313131	13 Rue Nationale 59000 Lille France	(50.63, 3.05)
3	La poste nationale	3631	13 r. nationale 59000 lille	(50.63, 3.05)

Table 1. Nomao spot partial description.

For data deduplication, we define an example as a comparison between two spots. Comparison techniques depend on the data type. For the geolocalization points, a geographical distance is used. For other values, the following string comparison functions are used: levenshtein, trigram, difference, inclusion, equality. Further details related to

string comparison functions like levenshtein or q-gram can be found in [8]. And all details about this data can be found on the challenge website `http://www.nomao.com/labs/challenge`.

As a consequence, a single example is defined by 118 comparison features. Its name is composed of the names of the spots that are compared, separated by a sharp (#), as shown in first column of table 2. In addition, a specific label is added corresponding to the final decision of data deduplication. A label value is +1 if the concerned spots must be merged, and -1 if they do not refer to the same place.

Considering examples described in table 1, the corresponding (partial) examples for data deduplication are provided on table 2. We assume there that an expert has qualified spots 1 and 2 as being distinct, as well as spots 1 and 3 (label -1), but spots 2 and 3 as being the same (label +1).

ID1#ID2	trigram(Name)	levenshtein(Phone)	levenshtein(Address)	distance(GPS)	label
1#2	1	0.3	0.78	0.99	-1
1#3	0.47	1	0.52	0.99	-1
2#3	0.47	0.3	0.74	1	+1

Table 2. Data deduplication examples.

2.3 Data distribution

In standard supervised learning, it is commonly assumed that the samples used for training follows the same probability distribution as the test samples. However, this assumption is not always satisfied in practice [9]. Dataset shift is present in most practical applications, for reasons ranging from the bias introduced by experimental design to the irreproducibility of the testing conditions at training time. The three main topics covered by this domain are (i) domain adaptation / transfer learning; (ii) covariate shift adaptation and (iii) multi-task learning.

A very simple way to analyze the difference between two distributions is to use a robust[5] classifier. The examples of both distributions are described by the same explanatory variables. A target variable is added on each distribution where its value is '+1' for the first distribution and '-1' for the second distribution. If the classifier is able to separate the two distributions, then its performance is an indication on the distance between the two distributions, and the variable importance provides this indication explanatory variable per explanatory variable.

This experiment has been conducted using the MODL approach which is a model selection method for classification and regression, that have no last recourse to cross-validation, yet performed well in recent benchmarks. Such methods have been recently extended to the less studied problem of rank regression. The methods used are Bayesian in spirit, but make use of original data-dependent priors [10].

[5] robust in the sense that it has strong regularization term.

Table 3 gives the 5 variables which are the more different between the train/test datasets and train/unlabeled datasets used in this challenge. An averaging of selective naive Bayes classifiers [11] obtains an Area Under the ROC curve (AUC) of 0.954 and 0.996 respectively when trained to discriminate the train/test distributions and the train/unlabeled distributions. These values, from our experience, indicate a strong difference between the various distributions. So an active strategy using a semi-supervised approach should be very interesting to be tested.

	Train / Test	Train / Unlabeled
Var1	phone_diff	street_number_diff
Var2	phone_levenshtein	street_number_levenshtein
Var3	phone_trigram	street_number_trigram
Var4	street_number_trigram	street_number_equality
Var5	geocode_coordinates	phone_levenshtein

Table 3. The 5 more important variables (sorted in descending order of relevance) to discriminate the distributions.

3 Initial in-house experiments

We report here the initial results of various active learning approaches that have been tested on Nomao data. They are all based on the use of *boosting* machine learning algorithm [12], and the selection of examples closest to the margin returned by the defined weak learners. Thus the active learning methods focus on examples that maximize the uncertainty about their label [13].

The first boosting algorithm that has been used is the classical *boosting of stumps* [14]. Then three methods for selecting examples have been considered:

1. one exploring the examples space, by selecting examples at *random*;
2. one exploiting fully the information coming from the boosting algorithm by selecting the examples closest to the *margin*;
3. and a last one mixing this exploitation of boosting with a bit of exploration of the examples space by using a random selection, weighted by the inverse distance to the margin: this approach will be called *wmargin*.

Hence, the process to get new examples to be labeled has the following steps:

1. learn on training data using the boosting of stumps with 100 trials (boosting steps);
2. use the model learnt to predict on unlabeled data (UD);
3. for the *margin* approach: select the examples closest to the margin;
4. for the *wmargin* approach: pick randomly the examples with their probability of being selected proportional to their associated distance to the margin.

As explained before, an initial (training) dataset had first been formed *by hand* that contained 29,104 examples. This dataset was cut randomly into 2 parts, in order to simulate the random selection[6]. Then we got our 2 datasets created using active learning. So finally we got the following 4 datasets:

1. *init* is the main one that contains 28,130 examples;
2. *rand* is the next one that picked randomly 974 examples;
3. *marg* contains 917 examples closest to the boosting margin;
4. *wmarg* contains 964 examples selected at random weighted by the distance to the margin.

3.1 Boosting stumps

Figure 2 shows how the testing phase is organized to evaluate the interest of those *active datasets*.

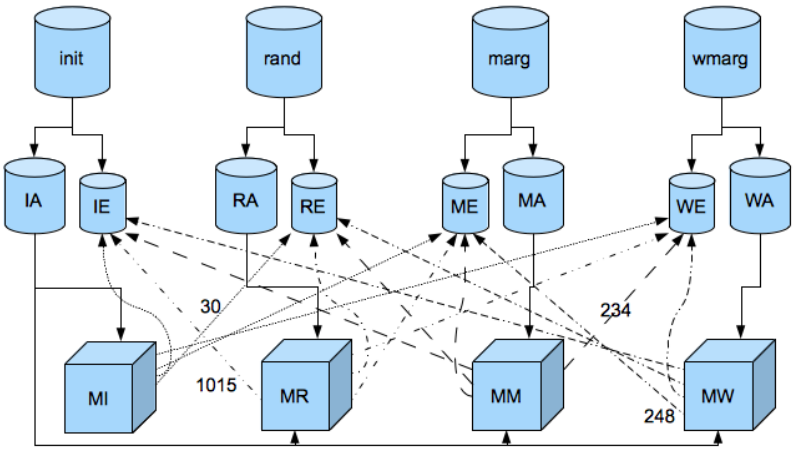

Fig. 2. Testing the interest of active datasets (e.g. 1015 examples of dataset *init* are misclassified by model *MR*).

Each dataset is cut into 10 parts to perform Cross Validation. For each test, 10 runs are thus performed, 9/10th of the data being used for training (*A*) and 1/10th for testing (*E*). Thus, in figure 2, *IA* corresponds to a training subset of the *init* dataset, *IE* to a test subset of *init*, *ME* a test subset of the *marg* dataset, and so on. Each active dataset (*rand*, *marg* or *wmarg*) is then associated to the initial one (*init*) in order to learn a model (called *MR*, *MM* or *MW*) using the boosting of stumps with 100 trials. A reference model *MI* based only on the initial dataset has also been computed. As a result, predictions are computed on all test datasets (*IE*, *RE*, *ME* and *WE*). Table 4 shows the results that have been obtained by that process.

[6] This way of selecting random examples will be improved in the next set of experiments described in the next section.

test / train	init (28,130)	rand (974)	marg (917)	wmarg (964)	error
initial only (reference)	**1006**	30	505	432	6.37%
+ *random* (explore)	1015	**29**	515	438	6.44%
+ *margin* (exploit)	1043	33	**243**	234	**5.01%**
+ *wmargin* (compromise)	1062	32	248	**230**	5.07%

Table 4. Number of misclassified examples with boosting of stumps, depending on the active learning method used to enrich the initial dataset for training.

We can thus observe that using active learning significantly improves the accuracy of the predictions, since the error rate decreases from more than 6% to roughly 5%.

The improvements are more significant on the examples that are supposed to be more difficult to predict, since we roughly divide by 2 the number of misclassified examples on the *marg* and *wmarg* datasets.

We can also note that all active approaches degrade the results on the initial dataset, and the approaches based on boosting even more. This means that adding to the training dataset too many difficult examples can affect the overall accuracy of the model.

These results are very interesting, even if we were surprised by the fact that picking random examples could decrease performance. Indeed, in these experiments, *random* is worse than *initial*, and *wmargin* worse than *margin*. This behavior indicates that the random strategy has (by chance) discovered a new "pattern" in the data which temporarily degrades the performances.

3.2 Boosting trees

Considering last results, we decided to carry out new tests with another boosting approach, based on the use of *decision trees* C5 [15] rather than stumps. Table 5 shows the results we got on the same datasets with this new algorithm, also run with 100 trials. The reader could also find a relevant reference in [16] for a tree-model strategy.

test / train	init (28,130)	rand (974)	marg (917)	wmarg (964)	error
initial only (reference)	466	10	251	266	3.20%
+ *random* (explore)	**444**	9	248	253	3.08%
+ *margin* (exploit)	496	11	**101**	129	2.38%
+ *wmargin* (compromise)	475	**8**	112	**96**	**2.23%**

Table 5. Number of examples misclassified by C5, depending on the active learning method used to enrich the initial dataset for training.

First of all we check here that the boosting of trees has better results than the boosting of stumps on Nomao data. Then we can observe that using active learning still significantly improves the accuracy of the predictions. But now the approaches based on the exploration of the examples space show improvements on the results. Indeed, the

number of misclassified examples is now lower with the *random* approach than with the *initial* one, and the *wmargin* has lower error than the *margin* one. The improvements are still more important on the examples selected closest to the margin.

Also, the best results are now obtained using the *wmargin* method. These results are more compliant to what is expected in active learning. Indeed, providing a compromise between exploration and exploitation has been shown to be important in active learning [17]. That way, the model is refined near to the decision boundaries, improving results on tricky examples, but the rest of the examples space is also explored in order to stay efficient on the rest of the dataset.

So the tests have been deepened in that direction: one new dataset has been generated: *wmarg5* contains 995 examples selected using the random selection weighted by the inverse distance to the margin provided by C5. We have also created a new random dataset in order to observe its effect on the error rate obtained on the initial dataset. That one contains 986 examples. So we will now use that one instead of the previous simulated one. Table 6 shows the results we got on those new datasets, using C5 algorithm again.

train \ test	init (29,104)	rand (986)	marg (917)	wmarg (964)	wmarg5 (995)	error
initial only	474	119	247	267	499	4.87%
+ *random*	**470**	**30**	221	224	445	4.22%
+ *margin*	494	49	**99**	129	403	3.56%
+ *wmargin*	493	37	103	**106**	372	3.37%
+ *wmargin5*	513	36	159	145	**198**	**3.19%**

Table 6. Number of examples misclassified by C5, depending on the active datasets used.

The reverse tests can also be conducted. Table 7 shows the results we got with C5 when using all data (*full*), or using all except those of one active dataset. In that case, the highest the error rate of an approach compared to the *full* one, the most useful the corresponding dataset is to the learning algorithm.

train \ test	init (29,104)	rand (986)	marg (917)	wmarg (964)	wmarg5 (995)	error
full	548	24	63	63	143	**2.55%**
no random	**571**	**29**	61	73	160	2.71%
no margin	540	26	**85**	74	160	2.68%
no wmargin	546	23	72	**85**	170	2.72%
no wmargin5	529	27	61	68	**218**	**2.74%**

Table 7. Number of examples misclassified by C5, depending on the active datasets NOT used.

These results are still more compliant to what we can expect from an active learning method. Each active dataset helps handling better its own kind of data, since *margin* is

the best approach to handle the *marg* dataset, *wmargin* the best for the *wmarg* dataset, and so on. Also, the random dataset now helps improving results on all data, including the initial one. At last, the overall performance of the system has increased significantly since we decreased its error rate to 2.55% using all data.

3.3 Discussion

This preliminary study has shown the difficulty of designing efficient active learning approaches in real-world applications.

We have shown that finding a good compromise between exploitation of information coming from the model used, and exploration of the examples space is not trivial. In particular, we have shown that a special care must be taken on finding active datasets that do not degrade the results of the model on the initial training data.

These results indicate also that the performances could be improved using better active learning methods or more adapted learning machines.

4 Nomao challenge

To deepen that research of performing Active Learning in Real-world Applications, a challenge has been organized, from Friday, June 1, to Friday, June 15, 2012 (see `http://www.nomao.com/labs/challenge`).

This section describes at first the protocol that has been set up for the challenge. We then present the baseline method used to assess the results of the participants. Finally, the last part of the section presents and discusses the results obtained.

4.1 Challenge protocol

999 new examples randomly selected were labeled by the Nomao expert in order to increase the size of the test dataset (TD) to 1,985 examples. The initial training dataset (ID) was still composed of 29,104 examples, and the unlabeled dataset (UD) of 100,000. The target variable was only provided to the participants for the training dataset.

During the *First Active Campaign*, participants could train a classifier using the training dataset (and the other datasets if they wanted to use semi-supervised method), and they returned us the predicted labels for the test dataset. These predicted labels allow us to measure the improvement obtained through active learning method. They also asked for N (set to 100) example labels belonging to the unlabeled dataset.

During the *Second Active Campaign*, they could train their classifier using the training dataset (and the other datasets if they wanted to use semi-supervised method), in addition to the N examples for which they asked the label. Then they returned us the predicted labels for the test dataset, while asking, again, for N (100) example labels belonging to the unlabeled dataset.

During the *Final Test Campaign*, they could train their classifier using the training dataset (and the other datasets if they wanted to use semi-supervised method), plus the $2 \times N$ examples for which they asked the label, and they returned us the predicted labels for the test dataset.

To obtain the final results we initially decided to rank the participants according to the improvement of the AUC. The goal was to have the best improvement thanks to active learning AND to beat the baseline model. But the participation of the challenge has been very low. Twelve competitors registered to download and analyze the data before the beginning of the first active campaign, but only two competitors entered in the first active campaign, and only one competitor achieved the complete process of the challenge.

Why did we observe this behavior? When writing this paper we do not have the answer, but we are investigating the reasons: problem too difficult, not enough advertizing, real-world issues? The ALRA workshop (`http://www.nomao.com/labs/alra`), to be held at the ECML-PKDD 2012 conference, will be the opportunity to discuss with the community about this point.

So at the end, the results have been analyzed in the light of the baseline model (revealed at the end of the challenge) and the last in-house experiment.

4.2 Baseline method

Planning the purchase of new examples (per batches) is a compromise between different steps which can be (i) a pre-selection [18]; (ii) a diversification [19]; (iii) the purchase of N labels; and (iv) the iteration evaluation [20]. These steps include the dilemma between exploration [21] and exploitation [22]. When the data are not purchased by batches the reader may find in [23] a relevant reference for a random selection baseline in applications.

The baseline method used to assess the performances of the competitors is based on the use of a single classifier, naive Bayes, and a simple active learning strategy. This baseline method is very straightforward and could easily be applied in the case where the labels have to be bought by batches.

The naive Bayes classifier comes from the software *Khiops*[7]. This is a naive Bayes classifier where each variable is weighted. The building phase of the weights of the variables is fully described in [11]. It includes two key steps: a step of variable selection (Section 3.5) and an averaging step (Section 6.2). The variable selection step allows the classifier to avoid unnecessary variables or explanatory variables unrelated to the classification problem. The averaging step allows weighting the variables. This classifier is a baseline classifier in the sense that it provides only one separation. More sophisticated classifiers, which incorporate several classifiers or linear separation, are known to be better than this baseline classifier to elaborate an active learning strategy [24].

The baseline active learning strategy used tries to combine exploration and exploitation. After training the naive Bayes classifier on the training data, the unlabeled dataset (UD) was ordered using the predicted probability for the class '+1'. The instances (X) to be labeled have then been chosen as described in figure 3 and correspond to a simple compromise between exploration and exploitation.

[7] `http://www.khiops.com`

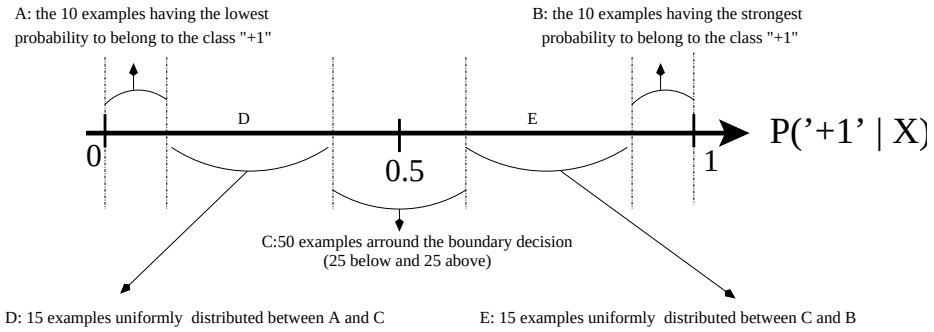

Fig. 3. Position of the labels asked versus P('+1'|X).

4.3 Results and discussion

During the challenge, the Nomao team has used the method described in section 3.2 that is based on the boosting of trees (*wmargin5*). The method employed by the winner, T. Sun, is described in [25]. Table 8 shows the AUC and error rate obtained by those 3 participants of the challenge on the test dataset (TD) and on their own active datasets (called AD).

Method	Baseline			Nomao			T. Sun		
Active phase	1	2	3	1	2	3	1	2	3
AUC on TD	0.9488	0.9786	0.9794	0.9807	0.9816	**0.9821**	0.9629	0.9631	0.9633
Error on TD	19.9%	9.6%	9.4%	12%	9%	7.5%	7.3%	**7.2%**	7.2%
Error on AD	37.8%	32.5%	$\emptyset$	32.5%	45%	$\emptyset$	24.4%	8.5%	$\emptyset$

Table 8. Results of participants on the test dataset (TD) and their own active datasets (AD).

T. Sun has clearly the best accuracy performances. In particular, he has shown very good results since the beginning, since he already reached 7.3% error even before the first active phase. But on the contrary, the other methods have shown high improvements of their approaches when helped with new active examples. The baseline method improved a lot with the first active phase (decreasing error from 19.9% to 9.6%), but much less on the second phase (from 9.6% to 9.4%), when Nomao improved its results more regularly (from 12% to 9% and then 7.5%).

All participants have selected difficult examples for the first active phase, since their errors were high on the corresponding examples (between 24.4% and 37.8% error). Then T. Sun selected examples on which he did much less error (8.5%), whereas Baseline and Nomao methods continued with high errors on their active examples (between 32.5% and 45%).

To check the overall difficulty of the targeted active examples, one may check the error of all methods on each dataset. Table 9 thus shows that the active data selected

by T. Sun and Nomao seem more difficult to predict than those chosen by the baseline method.

Method	Baseline	Nomao	T. Sun	Average
AD_1(Baseline)	$\emptyset$	8.5%	7.3%	8.6%
AD_2(Baseline)	$\emptyset$	8.6%	9.9%	
AD_1(Nomao)	37%	$\emptyset$	24.7%	21.6%
AD_2(Nomao)	18.6%	$\emptyset$	7%	
AD_1(T.Sun)	29%	38.4%	$\emptyset$	23.5%
AD_2(T.Sun)	16.7%	9.5%	$\emptyset$	

Table 9. Error rate obtained by every method when using their final version on the active datasets (AD) of other participants.

The Nomao expert reported that many examples were indeed difficult to qualify. Some of them were related to spots for whom the address was not precise (only the name of the town was available for instance). Pairs of spots could also refer to distinct shops in the same commercial centre, thus having the same address, and sometimes also the same phone number. The same case could arise with doctor's surgeries. Then post offices could also be tricky examples because their names and phone numbers were the same, so only the addresses were to make a difference.

Finally, table 10 shows the results of the 3 participants on all test datasets presented in this paper. We validate here that T. Sun has the best overall performances, and observe that even if Nomao outperformed the baseline method on the test set, Baseline had finally better overall results, since he got lowest error rates on difficult datasets such as *marg*, *wmarg* or *wmarg5*.

Method	Baseline	Nomao	T. Sun
test	9.4%	7.5%	7.2%
marg	22.9%	24%	17.9%
wmarg	21.3%	22.4%	16.5%
wmarg5	33.1%	45.8%	26.3%
total	19%	22%	**15%**

Table 10. Error obtained by every method when using their final version on all datasets.

5 Conclusion

The task that has been tackled during that challenge was especially difficult because the initial dataset had first been formed *by hand* by the Nomao expert. The distribution of the dataset was thus biased, so predicting labels on randomly selected examples (test dataset) has not been trivial, and we have shown that this has been even more difficult

when faced with examples selected near to the decision boundaries of the classifiers (active datasets). We face here a real-world situation.

Another aspect of the real-world anchor of this study concerns the initial process for selecting examples to be labeled. Datasets *marg*, *wmarg*, *wmarg5* and *rand* have been created sequentially, in order to develop the Nomao deduplication system as fast as possible. On the other side, datasets *baseline*, *nomao* and *tsun* have been generated in parallel, in a real research process. Therefore, even if they are both based on the same active learning paradigm, datasets *wmarg5* and *nomao* differ somehow.

Thanks to this study, the Nomao dataset has now grown from 29,104 to 34,465 examples, and the classifier is much more efficient in predicting labels, even for "tricky examples". Indeed, looking at the first result line of table 11, we can see that C5 has an error rate lower than 3% on the whole dataset.

train \ test	init (29,104)	rand (1,985)	marg (917)	wmarg (964)	wmarg5 (995)	baseline (163)	nomao (167)	tsun (170)	error
full	568	108	61	65	152	11	22	26	2.94%
no random	572	**115**	59	65	156	11	21	25	2.97%
no margin	547	108	**84**	**85**	163	11	24	27	3.04%
no wmargin	570	110	69	79	167	12	26	25	3.07%
no wmargin5	525	105	73	74	**269**	12	**27**	**28**	**3.23%**
no baseline	570	109	55	65	155	**13**	24	27	2.95%
no nomao	**577**	107	54	64	148	10	24	27	2.93%
no tsun	564	105	57	61	149	11	22	26	**2.89%**

Table 11. Final number of examples misclassified by C5, depending on the active datasets NOT used.

It can also be noticed that its error is much more important on the *wmarg5* dataset. This is especially obvious when *wmarg5* is not included in the training set but considered as a test set, since C5 then reaches an error rate of 269 / 995 = 27%. On the contrary, if this dataset is used, the error is significantly decreased from 3.23% to 2.94%. In fact, the active selection of examples using the *wmargin5* approach is here the best to improve C5 results.

rand, *marg* and *wmarg* datasets also improve the results of C5. *baseline* has not a significant influence. *nomao* seems the best to improve results on the *init* dataset, but its interest is not globally significant. And using *tsun* even decreases the precision of C5.

If *nomao* does not help that much improving C5, it can be because *wmarg5* already did the job. And if *tsun* is not helpful for C5, it can be because the active data requested by T. Sun was mainly adapted to the machine learning model he was using. In other words, we are probably observing here that **the relevance of the active learning process is model-dependent**.

Results reported in table 12 confirm that assumption since the active examples selected by *nomao* are indeed the most effective to improve C5's results. Then the *baseline* active examples lead to better results than when *tsun* active examples are used.

C5 (init)	C5 (init+nomao)	C5 (init+baseline)	C5 (init+tsun)
240	**148**	155	185

Table 12. Number of test examples (among 1,985) misclassified by C5, depending on the active datasets used.

To validate this assumption and deepen this research, we could now conduct this study again with the challenge winner's approach. We could thus at the same time improve Nomao's deduplication process and understand better how the context of use of a machine learning method must be taken into account when designing an Active Learning method for a Real-world Application.

Besides, since the initial data is a large proportion of the training data in the experiments, it would be interesting to see if the results would change for a much lower proportion.

To share that problem with the community, the whole labeled dataset has been delivered publicly to the UCI Machine Learning Repository [26].

6 Acknowledgement

We greatly thank Max Chevalier for his collaboration in the organization of this challenge, the corresponding ALRA workshop (`http://www.nomao.com/labs/alra`), and the writing of this paper.

References

1. White, R.: Motivation reconsidered: The concept of competence. Psychological Review **66** (1959) 297–333
2. Settles, B.: Active learning literature survey. Technical report, University of Wisconsin-Madison (2009)
3. Singh, A., Nowak, R., Ramanathan, P.: Active learning for adaptive mobile sensing networks. In: IPSN '06: Proceedings of the fifth international conference on Information processing in sensor networks, New York, NY, USA, ACM Press (2006) 60–68
4. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: Proc. 18th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA (2001) 441–448
5. Candillier, L.: Nomao : la recherche géolocalisée personnalisée. In Zighed, D.A., Venturini, G., eds.: 11ème Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances (EGC). Volume 1. (2011) 259–261
6. Sarawagi, S., Bhamidipaty, A.: Interactive deduplication using active learning. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2002) 269–278
7. Lemaire, V., Bondu, A., Clérot, F.: Purchase of data labels by batches: study of the impact on the planning of two active learning strategies. Technical report, Orange Labs (2007) `http://perso.rd.francetelecom.fr/lemaire/publis/iconip_2007_camera_ready.pdf`.
8. Navarro, G.: A guided tour to approximate string matching. ACM Comput. Surv. **33**(1) (March 2001) 31–88

9. Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D.: Dataset Shift in Machine Learning. The MIT Press (2009)
10. Guyon, I., Saffari, A., Dror, G., Cawley, G.: Model selection: Beyond the bayesian/frequentist divide. J. Mach. Learn. Res. **11** (March 2010) 61–87
11. Boullé, M.: Compression-based averaging of selective naive Bayes classifiers. Journal of Machine Learning Research **8** (2007) 1659–1685
12. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences **55**(1) (1997) 119–139
13. Wang, Z., Song, Y., Zha, C.: Efficient active learning with boosting. In: Proceedings of the 9th SIAM International Conference on Data Mining. (2009) 1232–1243
14. Torre, F., Faddoul, J.B., Chidlovskii, B., Gilleron, R.: Boosting multi-task weak learners with applications to textual and social data. In: 9th International Conference on Machine Learning and Applications (ICMLA). (2010) 367–372
15. Quinlan, R.: Bagging, boosting and c4.5. In: 13th National Conference on Artificial Intelligence. (1996) 725–730
16. Borisov, A.: Active batch learning with stochastic query by forest. JMLR: Workshop and Conference Proceedings (2010) in I. Guyon, G. Cawley, D; dDror, V. Lemaire and A. Statnikov editors.
17. Bondu, A., Lemaire, V., Boullé, M.: Exploration vs. exploitation in active learning : a bayesian approach. In: International Joint Conference on Neural Networks (IJCNN). (2010)
18. Gosselin, P.H., Cord, M.: Active learning techniques for user interactive systems : application to image retrieval. In: International Workshop on Machine Learning for MultiMedia (In conjonction with ICML). (2005)
19. Brinker, K.: Incorporating diversity in active learning with support vector machines. In: International Conference on Machine Learning (ICML). (2003) 59–66
20. Culver, M., Kun, D., Scott, S.: Active learning to maximize area under the roc curve. In: International Conference on Data Mining (ICDM). (2006)
21. Thrun, S.: Exploration in active learning. In: Handbook of Brain Science and Neural Networks. Michael Arbib (2007)
22. Osugi, T., Kun, D., Scott, S.: Balancing exploration and exploitation: A new algorithm for active machine learning. In: International Conference on Data Mining (ICDM). (2005)
23. Cawley, G.: Some baseline methods for the active learing challenge. JMLR: Workshop and Conference Proceedings **1** (2010) in I. Guyon, G. Cawley, D; dDror, V. Lemaire and A. Statnikov editors.
24. Xu, Z., Akella, R., Zhang, Y.: Incorporating diversity and density in active learning for relevance feedback. In: Proceedings of the 29th European conference on IR research. ECIR'07 (2007) 246–257
25. Sun, T., Zhou, J.: Batch-mode active learning by using misclassified data. In: Proc. of the ALRA Workshop (Active Learning in Real-world Applications), held at ECML-PKDD. (2012)
26. Frank, A., Asuncion, A.: UCI machine learning repository [http://archive.ics.uci.edu/ml] (2010)