

Sweetening the Dataset: Using Active Learning to Label Unlabelled Datasets

Rong Hu¹, Brian Mac Namee¹, and Sarah Jane Delany²

¹ School of Computing,
Dublin Institute of Technology, Dublin, Ireland

² Digital Media Centre,
Dublin Institute of Technology, Dublin, Ireland
rhu@comp.dit.ie, brian.macnamee@comp.dit.ie, sarahjane.delany@dmc.dit.ie

Abstract. Supervised machine learning approaches assume the existence of a large collection of manually labelled examples of the problem under consideration. However, in many cases such a collection does not exist and creating one is time consuming and expensive. This can be a barrier to the use of supervised learning in certain situations, particularly when the doubt as to whether the system will work or not makes the cost of creating a dataset unjustifiable. Active learning is a machine learning technique that has been used widely to create classification systems in the absence of large numbers of labelled examples, but that can also be used to create such collections. This paper will describe a system that uses active learning to label large collections of unlabelled data. We will show that the system can create an accurately labelled dataset approximately 10 times the size of the set of examples manually labelled by an expert. The experiments described are based on recipe data from the 1st Computer Cooking Contest to be held at ECCBR'08 and focus on identifying those recipes in the set that are desserts.

1 Introduction

The success or failure of machine learning based classification systems hangs on the datasets used to train them [1]. Without a good dataset it is impossible to build a quality classifier. A good dataset requires the existence of a large number of historical examples of the problem to be solved, which have been labelled with their solutions. However, manually generating such a collection of labelled examples is typically time-consuming and expensive (it usually involves expensive experts such as doctors or engineers) [2]. This can be a real barrier to the creation of classification systems for problems, as often the time or money is not available to generate a dataset.

For example, for the recent Irish referendum on the Treaty of Lisbon³ it would have been interesting to build classification systems that could determine whether online newspaper articles about the treaty were for or against it, or

³ www.europa.eu/lisbon_treaty/index_en.htm

whether they were partisan or non-partisan. However, training and evaluating such a classification system would have required a large collection (in the range of hundreds of documents) of labelled articles. While this collection could have been created, it would have taken someone a long time to label each article, particularly if we intended to try building a number of different systems speculatively in order to see what would be possible.

Fortunately, this is not an insurmountable problem. Creating labelled datasets can be approached using *active learning* [3], a machine learning technique that is used to build classifiers from collections of unlabelled data with the assistance of an oracle - typically a human expert. In this paper we use active learning to label unlabelled datasets (a less common use than for building classifiers) which can then be used to build classifiers (or for any other purpose). Our two, often competing, goals are to minimise the number of labels that are needed from the human expert while maintaining the quality of the labels applied by the system.

The paper will continue with a discussion of active learning in section 2. Section 3 will then present our active-learning-based labelling system, ALL, which uses case-based reasoning (CBR) at its heart. Section 4 will describe an experiment that has been performed to evaluate our system, discussing how the system has been used in an entry to the 1st Computer Cooking Contest to be held at ECCBR'08⁴ [4]. Finally, section 5 will conclude on what we have learned so far and discuss the directions in which we intend to take this work in the future.

2 Active Learning

The traditional approach to supervised learning is to train a classification model with a collection of previous examples of the problem in question, for which the correct classifications are known. This collection is known as a *labelled training set* and the approach has been referred to as *passive learning* [3] as the learner passively accepts the training examples to build a classification model. However, as was mentioned previously, for many problems generating a labelled training set is a time-consuming and expensive process. Active learning seeks to overcome this problem by iteratively training a classifier from an unlabelled training set using an oracle (typically a human expert) to only label those examples that are deemed most informative during the training process. The aim of this approach is to build quality classifiers using as few labelled training examples as possible. Active learning has been used in many applications including text classification [5, 6], outlier detection [7] and drug discovery [8].

The most common form of active learning is the *pool-based* approach [9, 10]. Pool-based active learning assumes that the learner has access to a large pool of unlabelled examples. The alternative, *stream-based* active learning [11, 12], assumes a constant stream of individual examples which must be dealt with. However, stream-based active learning will not be considered in this paper. For both approaches it is also assumed that for any examples the correct class label can be requested from an oracle.

⁴ www.computercookingcontest.net

In pool-based active learning the learning process begins by performing an initial attempt at building a classifier from a small set of examples that are labelled by the oracle. Using this initial classifier each member of the pool has associated with it a measure of how informative a label for that example would be to the training process. Those unlabelled examples for which labels are deemed most informative are then labelled by the oracle and removed from the pool. Following this, a new classifier is built using all of the examples labelled so far, and the process repeats as long as the oracle will continue to provide labels, or some other stopping criteria is reached - for example the classifier created has achieved a particular goal.

The predominant research issue in pool-based active learning is how the most informative examples should be selected from the pool for labelling. Selection strategies include *uncertainty sampling* [12], *version space reduction* [3] and *query-by-committee* (QBC) [11].

Uncertainty sampling was first proposed by Lewis and Gale [12] and has been widely used in active learning applications. Uncertainty sampling uses classifiers that can associate a certainty score with each of their classifications (ranking classifiers such as Naive Bayes, k -Nearest Neighbour and Support Vector Machines can do this). The certainty score, $P(C|e)$, indicates the certainty of the system that example e belongs to class C . Uncertainty scores typically fall into the range $(0, 1)$ where 0 indicates that the system is certain that the example does not belong to the class in question, and 1 indicates that it is completely certain that it does. At each iteration of the active learning process the certainty scores of each example are computed and those that are most uncertain (i.e. those with scores closest to 0.5) are selected for labelling. The philosophy behind this approach is that a better classifier can be built by reducing the uncertainty in the dataset. The advantages of the uncertainty sampling approach include its simplicity and fast execution speed.

Version space reduction based approaches select the examples which can most quickly reduce the size of the version space associated with the labelled examples. The idea is that the most informative examples are those which can eliminate whole portions of the version space. One of the most popular approaches within this group is support vector machine based active learning [3].

The *query-by-committee* (QBC) method creates a “committee” of classifier variants and classifies unlabelled examples with each committee member. Those examples with the biggest classification disagreement among the committee are then selected for labelling.

Other interesting extensions to active learning include Expectation-Maximization (EM) [13], Co-Testing [14], Co-Training [15], multi-label active learning [16], and batch mode active learning [17]. Visualisation is also particularly interesting in active learning [18].

While active learning is predominantly used for building classifiers, the approach can also be used for labelling unlabelled data sets. While this is a subtle difference (as labelling is involved as part of building a classifier) the end result (a labelled dataset) is much more flexible. For example, the labelled dataset can be

used to evaluate as well as train classifiers, or as an input to other systems outside of classification. There have been some interesting efforts in this direction in the past including labelling video streams [19] and labelling images [16]. However, this use of active learning has not received as much attention as the direct creation of classifiers. Section 3 will describe our approach to active-learning-based labelling which is intended to allow for the easy creation of labelled datasets, particularly from textual documents.

3 The ALL System

Our active-learning-based labelling (ALL) algorithm uses the ideas from pool-based active learning and uncertainty sampling. The dual goals of the algorithm are the creation of high-quality labelled datasets and the minimisation of the manual labelling effort. A flow diagram of the algorithm is shown in figure 1.

The algorithm starts with a large pool of unlabelled examples, each of which is to be labelled as belonging to one of two classes. A small number of examples of each class are initially selected from the pool (presently at random) and are manually labelled by the expert. These labelled examples form a case-base from which a ranking k -nearest neighbour (k -NN) classifier which uses distance-weighted voting [20] is built. This is the classifier components of our active learner.

Our choice of the k -NN classifier is informed by the fact that we feel it is uniquely suited to active learning. The reasons for this are that the introduction of new examples to the classifier simply involves adding them to the case-base, and that so much computation required for classification (e.g. the similarities between all cases) can be pre-computed. It is interesting to note that except for a small number of examples [21], k -NN (and more widely case-based reasoning (CBR) in general) has not been used widely in active learning research. This is something that we intend to pursue in the future.

The algorithm then proceeds by using the classifier created to classify each example remaining in the pool of unlabelled examples. As well as a class label each example, e , has associated with it a certainty score (as discussed in section 2), $certaintyScore(e)$. This is calculated as shown in Equation 1, where $posScore(e)$ is the sum of the similarities between the query example e and any nearest neighbours belonging to the positive class, and $negScore(e)$ is the sum of the similarities between e and any nearest neighbours belonging to the negative class. The pool is then ranked according to these scores.

$$certaintyScore(e) = \frac{posScore(e)}{posScore(e) + negScore(e)} \quad (1)$$

During the auto labelling process, the example that if the $certaintyScore$ is less than 0.5, then the label of it is 0; if greater than 0.5, the label is 1. The example in the pool with a $certaintyScore$ nearest to 0.5 is selected as the next example for labelling by the human expert and removed from the pool. The idea behind this is that this is the example that the system is currently most unsure

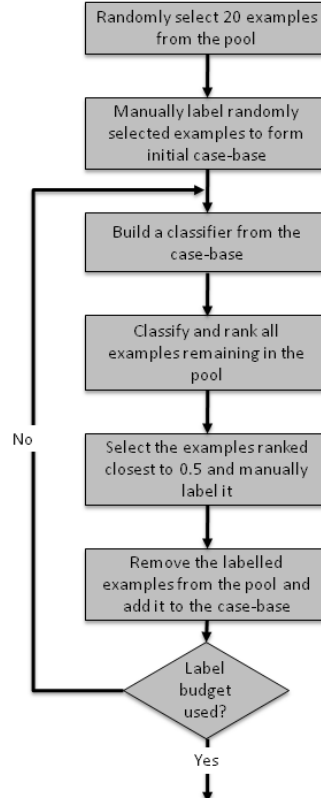


Fig. 1. The ALL algorithm.

about, and so labelling it will result in the most benefit to the system. After the example is labelled it is added into the k -NN case-base and the pool is re-labelled and re-ranked.

This re-building, re-classifying and re-ranking of the pool can make uncertainty-based active learning very computationally expensive. However, we feel that by using CBR, which allows so much of the computation to be pre-cached and calculated iteratively, much of this computational burden is overcome. In the k -NN classification model, the label of each query is defined by the labels of its k nearest neighbours. After adding a new case into the case-base, we simply need to compute the similarity between the newly added case and every example in the pool, as nothing else will have changed. Similarly, removing an example from the pool does not affect labelling in any way. So we can save the similarities between the pool and the labelled case-base as a similarity matrix and pre-cache this matrix. After each manual labelling, the case-base and the pool are changed but the similarity matrix is changed a little. By using the pre-cached matrix, we just need to update the matrix with very small part which can be achieved

by two operations including removing one column from the pool and calculating the similarity between the pool and the newly added example of case-base.

The process of selecting examples from the pool, labelling these examples, and re-ranking the pool continues until some stopping criterion is met. This stopping criterion establishes the balance between the number of labels provided by the user and the accuracy of the labels applied by the system. At present we use a simple stopping criterion that allows the human labeller to only provide a specified number of labels, a *label budget*. The number of labels to be allowed has been established through experimentation and currently is set to 10% of the number of examples in the pool. Section 4 will describe how this number was arrived upon and the results of the experiments which have been performed to evaluate our system.

4 Experimental Evaluation

The experiments undertaken to evaluate the performance of ALL were based on the dataset provided for the 1st Computer Cooking Contest which is to be held as part of ECCBR'08. The problem posed by the organisers of the contest is to build an automated system that can suggest a recipe to a user based on a set of requirements that they provide. The requirements which a user can provide include a set of ingredients, a cuisine type (e.g. Chinese or Mediterranean), and a particular course (e.g. starter or dessert). While a training set of XML recipes was provided as part of the challenge, it did not include labels to indicate cuisine type or course.

So, ALL was used to apply these labels so that they could be used in a retrieval system built as an entry to the contest [4]. The evaluation of ALL is based on its performance in this labelling task. In order to perform the evaluation a human expert first applied labels for one category (*desserts* or *non-desserts*) to every recipe in the dataset. The dataset comprised of 867 recipes of which 141 were labelled as desserts, and 726 were labelled as non-desserts. This set of labelled examples allowed us evaluate the accuracy of the labels created by the labelling system. Before considering the accuracy of the system in this task it is worth considering some of the details of the classifier used.

The similarity measure used to compare two recipes in the k -NN classifier used at the heart of the labeller was based on a weighted combination of the similarity between the ingredients used in the two recipes, and the similarity between their titles. The similarity between two ingredients is measured using the WordNet ontology [22]. The main food concept is parsed from the ingredient's textual description and matched to a valid concept in WordNet. The actual measure used to compare ingredients (Jiang and Conrath's path measure [23]) uses the shortest path length between the two concepts in the WordNet ontology and the density of the concepts along this path. Further details of this can be found in [4]. The title similarity measures the overlap similarity between the two title strings of the recipes. The overlap similarity is defined as the ratio of the number of same terms in both titles divided by the minimum number of terms

in both titles and the overlap similarity we used is implemented in jCOLIBRI2 [24]. In order to generate an overall similarity between two recipes, equal weights of 0.5 for ingredient similarity and 0.5 for title similarity were used.

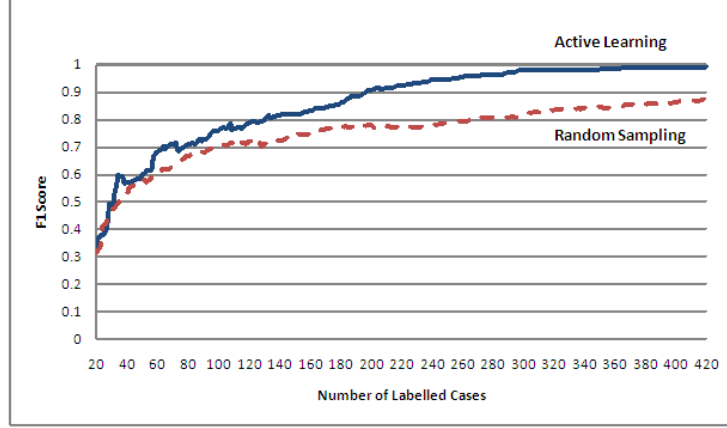


Fig. 2. F1 score comparison of active learner and random sampling

In order to perform the evaluation two experiments were conducted. In the first an initial set of 20 examples were selected at random from the pool and labelled by the expert. These were used as the initial case-base for the ALL algorithm, described in section 3, which was allowed run to a label-budget of 400 labels that is much bigger than the needed label-budget (that is 80 as described before) to achieve a comparable performance and give us a deeper insight into the ALL system. After each labelling the accuracy of the labels applied to each recipe was calculated, as was the $F1$ score [25]. Here the accuracy indicates the proportion of correct labelling made over a whole dataset including the initial case-base, and it measures the overall performance of the ALL system. The $F1$ score is defined as $F1 = 2 \times p \times r / (p + r)$, the harmonic mean of precision p and recall r . Precision is the ratio of correct labelling *desserts* (including the initial case-base and the manual labelling) divided by the total number of the system’s labelling that are labelled as *desserts* (including the initial case-base and the manual labelling). Recall is defined to be the ratio of correct labelling *desserts* (including the initial case-base and the manual labelling) by the system divided by the total number of “true” *desserts* in the dataset. In order to act as a comparison, a second experiment was run in which the examples to be labelled by the human expert were selected at random rather than using the certainty score. Again, this was allowed run to a total of 400 labels with accuracy and $F1$ score recorded after each labelling. For both of the experiments both the human labelled examples and the autonomously labelled examples were included in calculating accuracies and $F1$ scores. The $F1$ scores plotted against the number

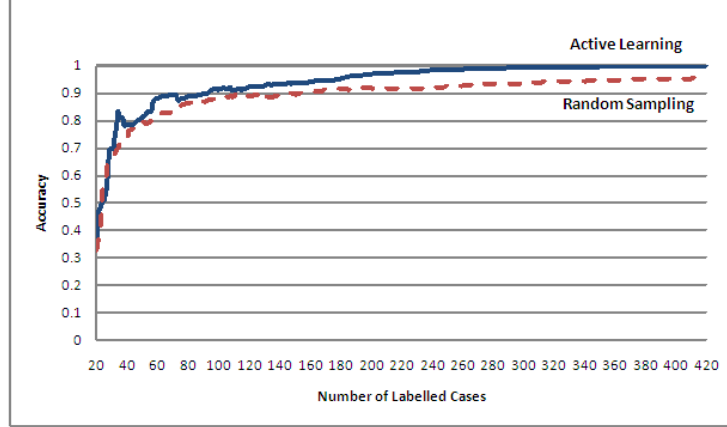


Fig. 3. Accuracy comparison of active learner and random sampling

of human labels are shown in figure 2, while the accuracies are shown in figure 3.

As can be seen from figures 2 and 3 the ALL system out-performs the system based on without replacement random sampling. In fact, after just 80 labels (approximately 10% of the examples beginning in the unlabelled pool) the accuracy achieved by ALL has reached over 90%, with an associated $F1$ score approaching 0.8. The 95% confidence interval for ALL for accuracy mean is [0.9312, 0.9486] while for random sampling is [0.8877, 0.9039]. The 95% confidence interval for ALL for $F1$ score is [0.8492, 0.8774] while for random sampling is [0.7423, 0.7626]. And paired t-tests show that the improvement of ALL in both $F1$ score and accuracy are statistically significant ($p < 0.0001$).

5 Conclusions & Future Work

This paper presented an approach to using active learning to label examples in unlabelled datasets. The purpose for this is to allow the creation of labelled datasets without the expense and time requirement imposed by complete manual labelling. In our experiments we have shown that reasonably accurate labels can be applied to a large dataset with a human labelling requirement of just 10% of the total number of examples. This would allow the creation of datasets that would not otherwise be feasible in terms of cost and time requirements. In particular we feel this is the case for textual datasets which are readily available, but extremely time consuming to label.

We expect that there are a number of improvements that we could make to this system, as presently it is based only on fairly simple active learning techniques. The addition of more sophisticated selection of the initial case-base (probably through clustering) and more interesting selection strategies should allow higher labelling accuracies be achieved with smaller numbers of manual

labels. Also, the work described here has focussed on binary labelling of the case-base, but we intend to extend this to multi-label situations in the near future.

However, most importantly we believe that the ALL labelling system is just the first step in a larger research agenda which will focus on dataset generation as a whole [26]. The easy availability of so much textual data on the web makes many classification tantalizingly possible if only datasets could be generated. Ultimately this will have to answer difficult questions such as how much data is enough, what kind of examples are required and how a dataset should be maintained.

References

1. Salzberg, S.: On comparing classifiers: A critique of current research and methods. *Data Mining and Knowledge Discovery* **1** (1999) 1–12
2. Patel, V.L., Groen, G.J.: Knowledge-based solution strategies in medical reasoning. *Cognitive Science* **10** (1986) 91–116
3. Tong, S.: Active Learning: Theory and applications. PhD thesis, Computer science department, Stanford University (2001)
4. Zhang, Q., Hu, R., Namee, B.M., Delany, S.J.: Back to the future: Knowledge light case base cookery. Technical report, Dublin Institute of Technology (2008)
5. McCallum, A.K., Nigam, K.: Employing em and pool-based active learning for text classification. In: *Proceedings of the Fifteenth International Conference on Machine Learning*, Morgan Kaufmann (1998)
6. Novak, B., Mladenić, D., Grobelnik, M. In: *Text Classification with Active Learning*. Springer Berlin Heidelberg (2006) 398–405
7. Abe, N., Zadrozny, B., Langford, J.: Outlier detection by active learning. In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. (2006) 504–509
8. Warmuth, M.K., Rätsch, G., Mathieson, M., Liao, J., Lemmen, C.: Active learning in the drug discovery process. In: *NIPS*. (2001) 1449–1456
9. Li, K.L., Li, K., Huang, H.K., Tian, S.F.: Active learning with simplified svms for spam categorization. In: *Proceedings of the First International Conference on Machine Learning and Cybernetics*. (2002) 1198–1202
10. Segal, R., Markowitz, T., Arnold, W.: Fast uncertainty sampling for labeling large e-mail corpora. In: *CEAS 2006: Conference on Email and Anti-Spam*. (2006)
11. H.S.Seung, M.Opper, H.Sompolinsky: Query by committee. In: *Proceedings of the Fifth Workshop on Computational Learning Theory*, San Mateo, CA, Morgan Kaufmann (1992) 287–294
12. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In Croft, W.B., van Rasbergen, C.J., eds.: *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, Dublin, IE, Springer Verlag, Heidelberg, DE (1994) 3–12
13. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.M.: Learning to classify text from labeled and unlabeled documents. In: *Proceedings of AAAI-98, 15th Conference of the American Association for Artificial Intelligence*, Madison, US, AAAI Press, Menlo Park, US (1998) 792–799
14. Muslea, I.A.: Active Learning with multiple views. PhD thesis, Faculty of the Graduate School, University of Southern California (2002)

15. Nigam, K., Ghani, R.: Understanding the behavior of co-training. In: KDD-2000 Workshop on Text Mining. (2000)
16. Li, X., Wang, L., Sung, E.: Multilabel svm active learning for image classification. In: Image Processing, 2004. ICIIP '04. 2004 International Conference on. Volume 4. (2004) 2207–2210 Vol. 4
17. 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, Edinburgh, Scotland, ACM (2006) 633–642
18. Turtinen, M., Pietikainen, M.: Labeling of textured data with co-training and active learning. In: Proc. the 4th International Workshop on Texture Analysis and Synthesis (Texture 2005), Beijing, China. (2005) 137–142
19. Yan, R., Yang, J., Hauptmann, A.: Automatically labeling video data using multi-class active learning. In: Proceedings of the Ninth International Conference on Computer Vision (ICCV'03). (2003) 516–523
20. Mitchell, T.: Machine Learning. McGraw Hill (1997)
21. Li, Y., Guo, L.: An active learning based tcm-knn algorithm for supervised network intrusion detection. Computers and Security **26** (2007) 459–467
22. Felllbaum, C.: Wordnet: An electronic lexical database. Cambridge (1998)
23. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. cmp-lg/9709008 (1997) In the Proceedings of ROCLING X, Taiwan, 1997.
24. Recio-García, J.A., Díaz-Agudo, B., González-Calero, P.: jcolibri2 tutorial (2007)
25. Jones, R., Owen, F.: Statistics. 4 edn. Financial Times/ Prentice Hall (1994)
26. Davidov, D., Gabrilovich, E., Markovitch, S.: Parameterized generation of labeled datasets for text categorization based on a hierarchical directory. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, Sheffield, United Kingdom, ACM (2004) 250–257

Acknowledgements

This material is based upon works supported by the Science Foundation Ireland under Grant No. 07/RFP/CMSF718.