Research Repository UCD

Title	Active learning for text classification with reusability
Authors(s)	Hu, Rong, MacNamee, Brian, Delany, Sarah Jane
Publication date	2016-03-01
Publication information	Hu, Rong, Brian MacNamee, and Sarah Jane Delany. “Active Learning for Text Classification with Reusability.” Elsevier, March 1, 2016. https://doi.org/10.1016/j.eswa.2015.10.003.
Publisher	Elsevier
Item record/more information	http://hdl.handle.net/10197/10851
Publisher's statement	This is the author’s version of a work that was accepted for publication in Expert Systems with Applications. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in Expert Systems with Applications (45, (2016)) https://doi.org/10.1016/j.eswa.2015.10.003
Publisher's version (DOI)	10.1016/j.eswa.2015.10.003

Downloaded 2026-05-01 23:36:33

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)

© Some rights reserved. For more information

Active Learning for Text Classification with Reusability ☆

Rong Hu[a], Brian Mac Namee[b], Sarah Jane Delany[a,*]

[a] *Applied Intelligence Research Centre, Dublin Institute of Technology, Ireland*
[b] *School of Computer Science, University College Dublin, Ireland*

Abstract

Where active learning with uncertainty sampling is used to generate training sets for classification applications, it is sensible to use the same type of classifier to select the most informative training examples as the type of classifier that will be used in the final classification application. There are scenarios, however, where this might not be possible, for example due to computational complexity. Such scenarios give rise to the reusability problem—are the training examples deemed most informative by one classifier type necessarily as informative for a different classifier types? This paper describes a novel exploration of the reusability problem in text classification scenarios. We measure the impact of using different classifier types in the active learning process and in the classification applications that use the results of active learning. We perform experiments on four different text classification problems, using the three classifier types most commonly used for text classification. We find that the reusability problem is a significant issue in text classification; that, if possible, the same classifier type should be used both in the application and during the active learning process; and that, if the ultimate classifier type is unknown, support vector machines should be used in active learning to maximise reusability.

Keywords:
active learning, machine learning, reusability problem, text classification

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

*Corresponding Author – ph: +353 1 402 4728, fax: +353 1 402 4985

Email addresses: amy.hur@gmail.com (Rong Hu), brian.macnamee@ucd.ie (Brian Mac Namee), sarahjane.delany@dit.ie (Sarah Jane Delany)

1. Introduction

Automated *text classification* (or *text categorisation*) (Yang, 1999) is the task of automatically assigning predefined categories to textual documents based on their contents. *Spam filtering* (Drucker et al., 1999) applications that sift through a user's incoming emails and identify those that are unsolicited, unwanted or inappropriate – those that are considered *spam* to the user – are a typical example. Another example is sentiment analysis (Pang et al., 2002) which aims to assist in the evaluation of documents – such as product reviews – by determining their overall sentiment (positive or negative). The relatively recent explosion of textual data from sources such as social network feeds and micro-blogging posts, on top of the already voluminous older sources such as SMS messages, online news articles, and blogs has made text classification an especially important problem within the machine learning community.

Text classification systems typically employ supervised learning approaches (Joachims, 1999; Yang & Liu, 1999) and, so, are reliant on the quality of the labelled historic datasets used to train them. Without a good dataset it is difficult to build an accurate classifier. Unfortunately, generating quality datasets usually requires manual labelling which is a time-consuming and, because experts are usually involved, expensive task. This can be a real barrier to the creation of classification systems but, fortunately, is not an insurmountable problem. *Active learning* (AL) (see Settles, 2009, for a review) is an iterative, semi-supervised learning process that can be used to build high-performance classifiers or labelled datasets by selecting only the *most informative* examples from a larger unlabelled dataset for labelling by an oracle (normally a human expert) and using these to train a classifier or infer the labels for the remainder of the unlabelled data. Previous work (Lewis & Catlett, 1994; Tong & Koller, 2001; Yu et al., 2008) has shown that active learning can reduce the number of labelled examples needed to build an accurate text classifier by as much as 90% and, so, makes feasible the prospect of building text classification systems that would otherwise require prohibitively expensive amounts of manual data labelling.

The key consideration in active learning is the design of *selection strategies* that select the most informative examples that will be presented to the oracle for labelling. *Uncertainty sampling* (Cohn et al., 1994, 1996; Lewis

& Gale, 1994; Tong & Koller, 2001) is the most commonly used selection strategy. When uncertainty sampling is used in active learning, each time new examples are labeled by the oracle a classifier is trained using these and all of the other examples labelled so far. This classifier is used to classify the remaining unlabelled examples and the certainties associated with these classifications is recorded. The examples with the lowest certainties associated with their classification are then presented for labelling and the process repeats until the maximum number of labels offered by the oracle is reached or some other stopping criteria has been met.

In many instances the classification algorithm used in the uncertainty sampling process is the same as the classification algorithm that will ultimately be used in the text categorisation system being constructed. Sometimes, however, the classification algorithm required for the final text categorisation system is not suitable for use in uncertainty sampling, or vice versa, and so the classification algorithms used will be different. There are a number of reasons that this scenario can arise including (i) that a classification algorithm might be too computationally expensive for use in an active learning selection strategy; (ii) that a text classification application might have particular classification algorithm requirements such as a capacity for explanation; or (iii) that the final form of the text classification system is not known at the time a labelled training set is created using active learning. This scenario gives rise to the *reusability problem* (Baldridge & Osborne, 2004; Tomanek et al., 2007): "*is a set of labelled examples that is deemed most informative using one classification algorithm necessarily informative for another classification algorithm?*"

While the reusability problem has been studied before – Tomanek & Olsson (2009) go so far as to suggest that the reusability problem is a barrier to the widespread adoption of active learning – there is no detailed, formal analysis of the problem in the context of text classification in the literature. This article presents such an analysis. Using the classification algorithms most commonly used in text classification, we consider the suitability of different pairs of classification algorithms used to select examples during active learning and then to perform classification in a resulting text classification application. We consider the following questions:

Q1: Does the reusability problem exist?

Q2: Does a homogeneous system in which the same classification algorithm

is used for both uncertainty sampling in active learning and the final text classification application always perform best?

Q3: Are there text classification algorithms that are particularly well suited to active learning selection regardless of the algorithms that will be used in final text classification applications?

Q4: Are there text classification algorithms that are particularly well suited to text classification applications built using data generated using active learning?

Based on the analyses of the questions listed above recommendations are made for the use of active learning in text classification. Ancillary issues such as computational efficiency are also considered.

The structure of this article is as follows. Section 2 first presents a comprehensive review of active learning and the reusability problem. Section 3 describes the methodology used in our experiments. Section 4 presents the evaluation performed to address the questions outlined above. Finally, Section 5 presents a set of recommendations based on this evaluation and discusses the directions in which the work will be taken in the future.

2. Related Work

Active learning first garnered serious research attention in the 1980s (Angluin, 1988) and since then has remained a vibrant research area. Active learning is widely used in situations where there are vast amounts of unlabelled data available (e.g. classification of astrophysical data (Schneider, 2009), image classification (Tong & Chang, 2001), natural language processing (Baldridge & Osborne, 2004) and text classification (Nigam & McCallum, 1998)) or where labelled training examples are expensive or time consuming to obtain (e.g. bioinformatics (Cebron & Berthold, 2006) or medical applications (Warmuth et al., 2003)). Although there are other approaches (active learning with membership queries (Angluin, 1988), and stream-based active learning (Freund et al., 1997; Chu et al., 2011; Lughofer, 2012b)), *pool-based* active learning is by far the most common approach to active learning (particularly when active learning is applied to text classification problems (Lewis & Gale, 1994; Nigam & McCallum, 1998; Tong & Koller, 2001)) and is the approach considered in this work.

Pool-based active learning assumes that the learner has access to a large pool of unlabelled examples from the beginning of the process. The goal is to build either an effective classifier or a fully labelled dataset (which will be most likely used at some point to train a classifier) by labelling only a small subset of the examples in the pool. This is achieved by selecting those examples from the unlabelled pool that are deemed to be *most informative* for labelling by an oracle (typically a human expert). Figure 1 shows a flow-chart of the active learning process. After an initialisation step, the informativeness of each example in the pool is ranked and those deemed most informative are selected for labelling by the oracle. The informativeness ranking of each unlabelled example is then updated and the process is repeated until some stopping criterion has been met.

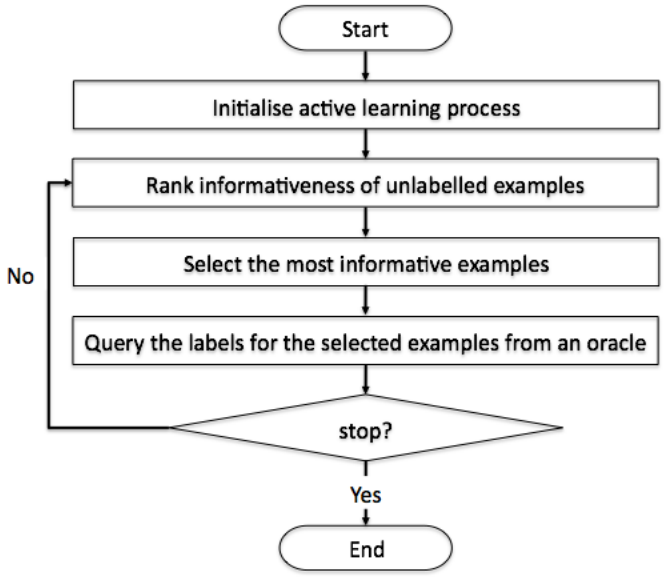


Figure 1: A flow-chart of the pool-based active learning process.

The key elements of the pool-based active learning process can be more formally modelled as a quintuple: $<\mathcal{S}, \mathcal{O}, \mathcal{L}, \mathcal{U}, \mathcal{SC}>$ (Baram et al., 2004). A small set of seeded examples, $\mathcal{L}$, that are labelled by an oracle, $\mathcal{O}$, is used to initialise a selection strategy, $\mathcal{S}$. The selection strategy first involves assigning each member of the unlabelled pool, $\mathcal{U}$, a value indicating how informative a label for that example would be to the active learning process. The examples for which labels are deemed most informative are then selected

for presentation to the oracle, $\mathcal{O}$, for labelling. The labelled examples are removed from the pool, $\mathcal{U}$, and added to the set of labelled examples, $\mathcal{L}$, and the informativeness values associated with each unlabelled example in $\mathcal{U}$ are updated. The process repeats as long as the oracle will continue to provide labels, or until some other stopping criterion, $\mathcal{SC}$, is reached. The final labelled set is then typically used to a build a classifier. This classifier itself can be the output of the active learning process or, alternatively, it can be used to label the remainder of the examples in the unlabelled pool with this completely labelled set then becoming the output of the process. An algorithm for a generic pool-based active learning system is shown in Algorithm 1.

Input: An initial training set $\mathcal{L}$, an unlabelled pool $\mathcal{U}$, a selection strategy $\mathcal{S}$, a stopping criterion $\mathcal{SC}$, a batch size b
Output: A labelled set or a classifier trained on the labelled set
while *SC is not met* **do**
  $Selected = \emptyset$;
  For each unlabelled example, assign a value to indicate its informativeness;
  Choose b most informative examples using $\mathcal{S}$;
  Add the b examples to $Selected$;
  Label each example $x_i \in Selected$;
  $\mathcal{L} = \mathcal{L} \cup Selected$, $\mathcal{U} = \mathcal{U}/Selected$;
end

Algorithm 1: A generic pool-based active learning algorithm.

There are three main questions in designing an active learning system:

1. How should the initial training set be selected?
2. What stopping criterion should be employed?
3. What is the best selection strategy to use?

Constructing the initial training set is often achieved by selecting a small number of examples from the unlabelled pool at random. This, however, ignores an opportunity to seed the process in a targeted way. It has been shown that this can be achieved by applying a clustering algorithm to the unlabelled pool and selecting the resulting cluster centres as the examples to seed the AL process (Zhu et al., 2008; Nguyen & Smeulders, 2004; Kang

et al., 2004; Lughofer, 2012a). Furthermore, it has been shown that deterministic clustering approaches, such as agglomerative hierarchical clustering (Voorhees, 1986) are most effective in this regard (Hu et al., 2010b).

Choosing when to stop the active learning process is most often simply dictated by the number of labels an oracle will provide (Novak et al., 2006) (often referred to as a *label budget*). There are, however, other approaches based on performance achieved on a hold-out test set (Campbell et al., 2000), although these suffer from the difficulty of getting labelled examples which necessitates the use of active learning in the first place. Some other promising approaches are based on the characteristics of a classifier (Schohn & Cohn, 2000; Ertekin et al., 2007; Schohn & Cohn, 2000) built using the examples labelled so far or its classifications on the unlabelled pool (Vlachos, 2008; Laws & Schütze, 2008; Zhu et al., 2010).

Active learning selection strategies can be categorised as *exploration-based* (also referred to as *unsupervised*) or *exploitation-based* (also referred to as *supervised*). Exploration-based selection strategies focus on properties of the unlabelled examples themselves and select examples for labelling based on these properties. For example, an exploration-based selection strategy might select examples from dense regions of the example space in order to label the most representative examples (Donmez et al., 2007; Dasgupta & Hsu, 2008); or alternatively might select examples distant from the current labelled set with the aim of sampling wider, potentially more interesting, areas of the example space (Baram et al., 2004); or use a combination of these two approaches (Hu et al., 2010a, 2009). The evaluations presented in this article focus on exploitation-based (or supervised) selection strategies and so a deeper discussion of exploration-based (or unsupervised) approaches is beyond the scope of this article (a good review can be found in (Hu et al., 2010a)).

Exploitation-based selection strategies build a model using those examples labelled by the oracle so far in the active learning process, and exploit this model to select the next batch of examples for labelling. *Uncertainty sampling* is the most widely used exploitation-based selection strategy in text classification (Lewis & Gale, 1994; Tong & Koller, 2001; Raghavan et al., 2006; Segal et al., 2006).

In uncertainty sampling a ranking classifier[1] (e.g. k-nearest neighbour,

[1] A ranking classifier is a classifier that outputs a score associated with each classification

naïve Bayes or support vector machine), trained using the examples labelled so far, is used to classify the unlabelled examples in the pool. The output of the ranking classifier is then used as a measure of the *uncertainty* of each classification, and those examples for which classifications are least certain are selected for labelling by the oracle. By focusing on uncertain classifications uncertainty sampling focuses on labelling examples near the current classification boundary so as to fine-tune this boundary.

While uncertainty sampling is the most commonly used exploitation-based active learning selection strategy, particularly for text classification, there are other useful approaches. *Query-by-committee* (QBC) (H.S.Seung et al., 1992) selects examples based on disagreement within a committee of classifiers, while *the estimated error reduction framework* (Roy & McCallum, 2001) estimates the expected future error due to the labelling of some unlabelled example and then selects the example that generates the lowest expected error. The evaluations presented in this work all use uncertainty sampling.

Before leaving the discussion of selection strategies it is worth noting that exploitation-based and exploration-based approaches have been combined into hybrid approaches that seek to benefit from the best aspects of both (e.g. (Xu et al., 2003, 2007; Cebron & Berthold, 2008)).

It is the use of a classifier by exploitation-based selection strategies in active learning that gives rise to the *reusability problem*. Tomanek (2010) uses the term *foreign selection* to refer to a scenario in which a set of examples, labelled using an active learning process that uses a classifier C_i, is used to train another classifier C_j where $C_i \neq C_j$. Tomanek (2010) refers to C_i as a *selector* and C_j as a *consumer*. Foreign selection is contrasted with *self selection* in which $C_i = C_j$.

Foreign selection may happen in the scenario where a cheap, efficient classifier is required in the active learning selection strategy, but the resulting labelled examples are used to train another, more computationally expensive classifier. Foreign selection may also arise when the type of classifier to be trained on the output of the active learning process is unknown at the time that the active learning process takes place. In these terms we can state the

that it makes. For binary classification problems, like those studied in this article, these scores are typically in the range $[0, 1]$ where 0 indicates an almost certain classification of the negative class and 1 indicates an almost certain classification of the positive class. Scores in the region of 0.5 indicate a large degree of uncertainty about a classification.

reusability problem as (Baldridge & Osborne, 2004; Tomanek et al., 2007): will the examples deemed most informative for a selector classifier be the best examples to ultimately train a consumer classifier? The reusability problem has been cited as one of the main barriers to the widespread adoption of active learning (Tomanek & Olsson, 2009).

One of the earliest works in reusability (Lewis & Catlett, 1994) described a scenario in which a highly efficient probabilistic classifier selects examples for training a decision tree classifier. Experimental results showed that labelled training examples from the first classifier could be used to successfully train the second classifier and achieve better performance than if the training examples were selected at random. Later work, however, has contradicted this result. For example, Baldridge & Osborne (2004) used an ensemble-based active learning method for creating labelled data for head-driven phrase structure grammar (HPSG) parse selection, and showed that sample reuse was worse than random sampling. These types of contradictory findings have renewed interest in the reusability problem.

The most comprehensive work to date on reusability is by Tomanek & Morik (2010), who systematically investigated the reusability problem for uncertainty sampling on general classification problems, as well as on named entity recognition problems (an important task in natural language processing). The following conclusions were drawn:

- R1: Labelled examples obtained by active learning with a particular classifier are generally reusable by another classifier.
- R2: For a particular classifier C_j, that is trained on a set of labelled examples from an active learning process based on a classifier C_i, the performance of self-selection, where $C_i = C_j$, is occasionally bettered by foreign-selection, where $C_i \neq C_j$.
- R3: In the case of foreign-selection, no conclusion can be drawn about combinations of selectors and consumers that generally work well together.
- R4: A high degree of model similarity between the selector classifier and the consumer classifier often leads to high reusability, but low model similarity does not necessarily imply a low level of reusability.
- R5: Neither the distributional similarity nor the similarity of feature ranking of labelled examples can explain reusability.

In this work we will seek to build on these conclusions and make concrete recommendations with regard to the classifier types that should be used as selectors and consumers in active learning based text classification scenarios. The following section will describe the experimental methodology we have used to examine the performance of a range of selector and consumer classifier pairs on a selection of text classification problems.

3. Experimental Methodology

This section will describe the experimental methodology used in the evaluation experiments described in this article. In particular we will describe the evaluation measures used, the datasets used and the experimental setup.

3.1. Evaluation Measures

There are a number of different quantitative approaches to evaluating active learning (see Schütze et al. (2006) for a good overview). In classification settings fully labelled datasets are usually used to simulate the active learning process in order to perform evaluations, and performance may be measured at the completion of the process using measures common in general classification research, e.g. generalisation accuracy (or error) (Andrew et al., 2005), F-measure (Ando & Zhang, 2005) or *precision-recall break-even point* (PRBEP) (Tong & Koller, 2001).

Fully evaluating an active learning system, however, requires a more detailed analysis of the process of labelling a dataset, rather than simply the end result. For this reason, *learning curves* (for an example see Figure 2(a)) are widely used to monitor the progress of the active learning process in terms of classifier performance. Usually a learning curve is plotted with the number of labels given by the oracle on the *x-axis* and a performance measure on the *y-axis* (see Tong & Koller, 2001; Schohn & Cohn, 2000; Roy & McCallum, 2001, for examples). From the learning curve a global assessment score, such as the *area under the learning curve* (AULC), for an active learning approach can be calculated. One example of this can be seen in (Guyon et al., 2010) where the learning curve plots the *area under the ROC curve* (AUC) on the *y-axis*, as a function of the number of labels queried so far in the process on the *x-axis*. Other approaches include *deficiency* (Baram et al., 2004) and *efficiency* (Raghavan et al., 2006) which are defined in terms of classification performance (accuracy and F1 measure respectively) to measure the learning rate of an active learner.

Time performance has also been used to measure the efficiency of active learning algorithms, as timely performance is important in making active learning applications practically feasible. Hoi et al. (2009) measures the average CPU time needed to label one example. Segal et al. (2006) examined the CPU time needed to achieve some particular goal (e.g., accuracy). Probst & Ghani (2007) reported the time needed to wait between active learning interactions.

Tomanek & Morik (2010) proposed a measure of REUsability (REU) which measures the percentage decrease of the active learning self-selection sampling efficiency by active learning foreign-selection relative to the baseline random sampling scenario. REU is calculated based on AULC scores in a learning stage starting from a number of a labelled examples and ending with a number of b examples on the *x-axis* (an interval $[a, b]$). In order to make the REU score meaningful, however, two assumptions are necessary. The first is that self-selection would constitute the upper efficiency bound for foreign-selection. The second is that the baseline selection strategy would constitute the lower bound for foreign-selection. Unfortunately, these two assumptions do not always hold in practical applications and, furthermore, there is a problem in using random sampling as the baseline since it introduces an element of non-determinism into experiments.

For these reasons we do not use REU in our experiments. Instead, we directly use the area under the learning curve (AULC) to evaluate active learning approaches, where our learning curves plot the generalisation accuracy on a hold-out test set of a classifier built using the examples labelled so far. We also compare performances based on the processing time required for each selector.

3.2. Datasets

Seven balanced, textual, binary classification datasets are used in our evaluations. These include a spam dataset that contains emails classified as spam and non-spam (Delany et al., 2005); four datasets in which newsgroup posts are classified into one of two distinct topics derived from the widely used 20-Newsgroup collection[2]; a dataset from the Reuters collection[3] in which articles on two different subjects are distinguished; and a dataset derived

[2]Available from: http://people.csail.mit.edu/jrennie/20Newsgroups/

[3]Available from: http://www.daviddlewis.com/resources/testcollections/reuters21578/

from the RCV1 collection (Rose et al., 2002) which also distinguishes between different topics. Each document is tokenised and represented as a vector of words where the feature values represent the frequency of occurrence of the word. Before tokenisation, the words in each document are stemmed using Porter stemming (Porter, 1980) and stop-words are removed using a common English stop-words list. In addition, document frequency reduction is used where words that occur in less than three documents are removed. For classifiers other than naïve Bayes (i.e. the SVM classifier or the k-NN classifier) the term vectors are normalised to unit length. The properties of each dataset are shown in Table 1. This table also indicates the topics used in the datasets derived from the 20-Newsgroup, Reuters and RCV1 collections.

Table 1: Details of the datasets used in our evaluation experiments.

Dataset	**topics**	**#examples**	**#features**
20NG-WinXwin	comp.os.ms-windows.misc vs. comp.windows.x	$1,945$	$16,633$
20NG-Comp	comp.sys.ibm.pc.hardware vs. comp.sys.mac.hardware	$1,943$	$13,808$
20NG-Talk	talk.religion.misc vs. alt.atheism	$1,427$	$14,160$
20NG-Vehicle	rec.autos vs. rec.motorcycles	$1,984$	$14,470$
Reuters	acq vs. earn	$1,804$	$7,404$
RCV1	g151 vs. g158	$2,000$	$10,928$
Spam	spam vs. non-spam	$1,000$	$29,985$

3.3. Experimental Setup

The experiments performed essentially compare the performance of different selector and consumer pairs when used in active learning on each of the seven datasets described in the previous section. To ensure the validity of our results each single experiment run (i.e. a particular selector/consumer pair and dataset) is repeated five times using different randomly selected hold-out sets (a form of k-fold cross validation) and average results are reported. On each run the examples in the hold-out test set are not used in the active learning process itself.

In all of the experiments the same active learning process is used. For each pool, as in Hu et al. (2010b), an initial training set containing 10 seed examples is selected using agglomerative hierarchical clustering (Voorhees,

1986). The same initial training set is used in each experiment which uses that pool. In each run of the experiments, a selector is used to determine the examples to select from the pool and present for labelling, then a consumer is trained on the resulting labelled examples and tested on the hold-out test set. The predictions made by the consumer are compared with the actual labels and the accuracy is recorded. This process is repeated until a label budget of 500 labels expires. As the datasets used in the evaluations are fully labelled, the labelling process can be simulated without the need for an actual human oracle. Using the accuracy recorded after each manual labelling, a learning curve is constructed to plot the accuracy on the y-axis as a function of the number of labels provided on the x-axis (for an example see Figure 2(a)). The *area under the learning curve* (AULC) score is then used to measure the overall performance.

The classifiers used in this study are: a k-nearest neighbour (k-NN) classifier (where k is always set to five) that uses cosine similarity and distance weighted majority voting; a support vector machine (SVM) classifier using a linear kernel; and a multinomial naïve Bayes classifier. These three classifiers were chosen as they are the most common approaches used for text classification tasks. Parameters for all classifiers (e.g. the value of k for the k-nearest neighbour classifier) were kept consistent across all experiments described in this article so as to reduce variability across experiments, and because (although it is possible) it is very difficult to reliably optimise parameters for the selector classifiers since they are working predominantly with unlabelled data.

As in Tomanek & Morik (2010), when the naïve Bayes classifier is used as a selector the uncertainty of an example x is defined as:

$$\text{uncertainty}(x) \quad = 1 - |p(x \in \text{class1}) - p(x \in \text{class2})| \tag{1}$$

where $p(x \in \text{class1})$ and $p(x \in \text{class2})$ are the outputs of the naïve Bayes classifier.

For the SVM selector, following previous work by Tong & Koller (2001), the uncertainty of an example is defined as its distance from the separating hyperplane. In other words, examples that are closer to the hyperplane are more uncertain than those that are further away.

For the k-NN selector, following work presented by Hu et al. (2008), a

ranking score $r(x)$ is defined as:

$$r(x) = \frac{\sum_{1 \leq i \leq k,\ NN_i(x) \in class1} Sim(x, NN_i(x))}{\sum_{1 \leq i \leq k} Sim(x, NN_i(x))} \tag{2}$$

where $NN_i(x)$ is the i_th nearest neighbour of x, and $Sim(x, y)$ is the cosine similarity between examples x and y. Examples with ranking scores closest to 0.5 (complete uncertainty) are selected for labelling first.

4. Evaluation

In Section 1 we summarised the focus of this article in a list of 4 key research questions (Q1, Q2, Q3, and Q4). The evaluations presented in this section are designed to answer these questions. Furthermore, questions Q3 and Q4 will be further broken down so as to identify the best classifier used to select examples to label through active learning (i.e. the best *selector*) and the best classifier to be trained on a labelled training set which has been obtained using active learning methods (i.e. the best *consumer*. More precisely, the following four questions will be answered:

Q3a: If the consumer is known, which selector should be used?

Q3b: If the consumer is unknown, which selector should be used?

Q4a: If the selector is known, which consumer should be used?

Q4b: If the selector is unknown, which consumer should be used?

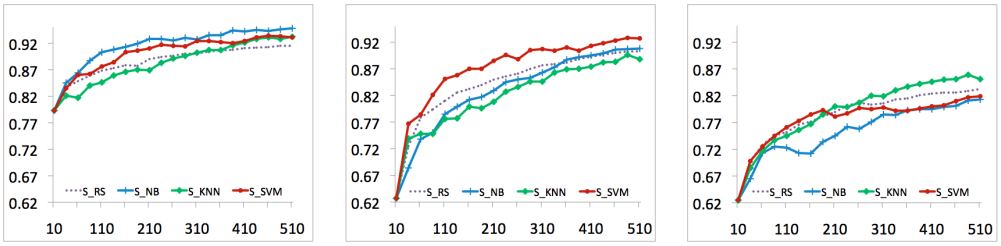
When trying to identify the best selector, i.e. to answer Q3a and Q3b, the terms *self-selection* and *foreign-selection* are used, following Tomanek & Morik (2010). Self-selection refers to the scenario where the same type of classifier is used as the selector during the active learning process as that used to consume the resulting labelled dataset. In contrast, foreign-selection refers to a scenario where the classifier used as the selector during active learning and the classifier used to consume the resulting dataset are different. When trying to identify the best consumer, i.e. to answer Q4a and Q4b, we use the terms *self-reuse* and *foreign-reuse* which have similar meanings to self-selection and foreign-selection.

4.1. Does the Reusability Problem Exist?

Figures 2 to 8 show the learning curves for the naïve Bayes, support vector machine and k-nearest neighbour consumers paired with the naïve Bayes, support vector machine, k-nearest neighbour and random selectors on each of the 7 datasets used in this study. The random sampling selector (S_RS), which randomly selects examples to label, is included as a baseline measure. We run the random sampling five times and report the averaged accuracy of the five runs. Note that in all learning curves the vertical axes are magnified to improve legibility.

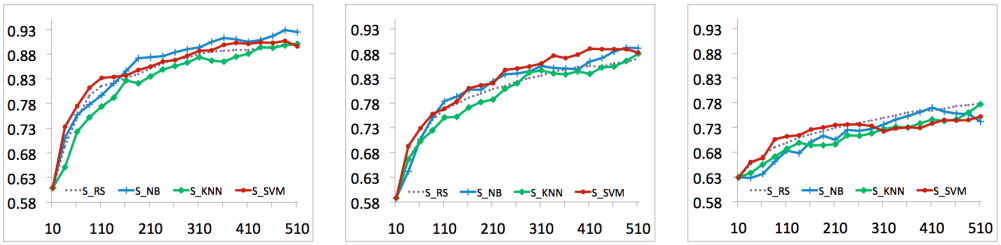
As can be seen from these figures, for both the naïve Bayes and SVM consumers, the learning curve of self-selection dominates those of foreign-selection and random selection for all datasets. This indicates that self-selection performs better than foreign-selection and random selection for these consumer classifiers. For the k-NN consumer, self-selection outperforms foreign-selection on four out of seven datasets (20NG-WinXwin, 20NG-Vehicle, Reuters-1804 and RCV1-2000) which confirms one of the findings from Tomanek & Morik that self-selection is occasionally outperformed by foreign-selection.

In order to analyse these results in more detail, we calculated AULC scores for all experiments. Table 2 shows the AULC scores for the four selectors when different consumers are trained on the active learning selected examples on the seven datasets. The AULC score of the best selector is highlighted in bold. The letters in parentheses indicate whether the best selector for a particular consumer is a situation of self-selection (s) or foreign-selection (f) (the use of random selection is considered an instance of foreign selection).



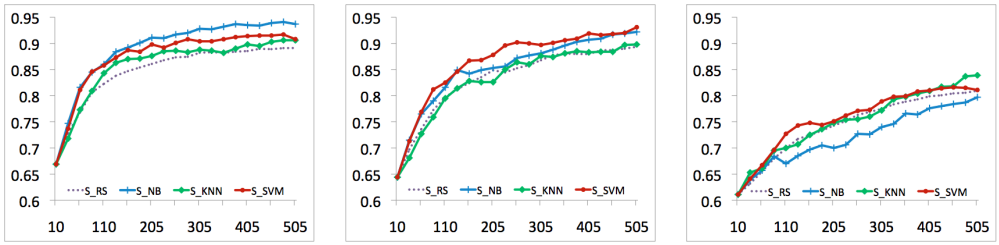
(a) NB consumer (b) SVM consumer (c) k-NN consumer

Figure 2: Learning curves of three consumers with different selectors (as shown in legend) on the 20NG-WinXwin dataset.



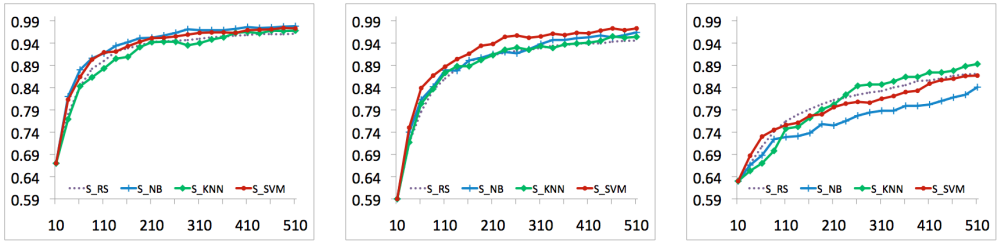
(a) NB consumer (b) SVM consumer (c) k-NN consumer

Figure 3: Learning curves of three consumers with different selectors (as shown in legend) on the 20NG-Comp dataset.



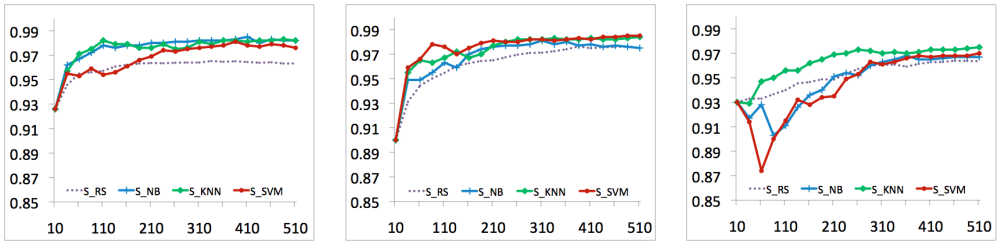
(a) NB consumer (b) SVM consumer (c) k-NN consumer

Figure 4: Learning curves of three consumers with different selectors (as shown in legend) on the 20NG-Talk dataset.



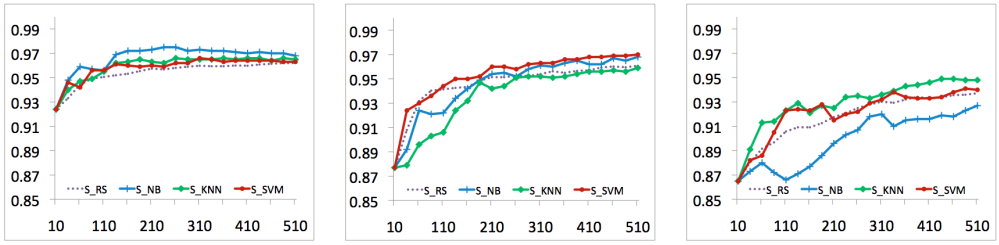
(a) NB consumer (b) SVM consumer (c) k-NN consumer

Figure 5: Learning curves of three consumers with different selectors (as shown in legend) on the 20NG-Vehicle dataset.



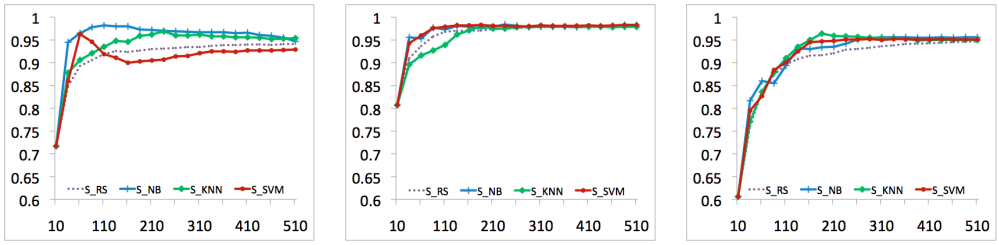
(a) NB consumer (b) SVM consumer (c) k-NN consumer

Figure 6: Learning curves of three consumers with different selectors (as shown in legend) on the Reuters-1804 dataset.



(a) NB consumer (b) SVM consumer (c) k-NN consumer

Figure 7: Learning curves of three consumers with different selectors (as shown in legend) on the RCV1-2000 dataset.



(a) NB consumer (b) SVM consumer (c) k-NN consumer

Figure 8: Learning curves of three consumers with different selectors (as shown in legend) on the Spam-1000 dataset.

Table 2: Results for identifying the best **selector**. The AULC score of the best selector is highlighted in bold, and letters in parentheses indicate that the best selector is in a situation of self-selection (s) or foreign-selection (f) (the use of random selection is considered an instance of foreign selection). Cases in which a selection strategy is outperformed by random selection are highlighted using italics.

		NB Selector	**k-NN Selector**	**SVM Selector**	**RS Selector**
NB Consumer	**20NG-WinXwin**	**458.5(s)**	*441.4*	451.3	443.7
	20NG-Comp	**429.6(s)**	*413.6*	426.6	421.9
	20NG-Talk	**447.3(s)**	430.6	439.5	425.4
	20NG-Vehicle	**470.3(s)**	*459.2*	467.1	461.2
	Reuters-1804	**488.8(s)**	488.3	484.2	480.5
	RCV1-2000	**483.8(s)**	480.1	479.4	477.3
	Spam-1000	**481.5(s)**	472.0	*458.6*	460.8
SVM Consumer	**20NG-WinXwin**	*416.6*	*409.9*	**437.9(s)**	423.8
	20NG-Comp	407.8	*397.1*	**413.1(s)**	402.0
	20NG-Talk	427.2	*417.1*	**434.0(s)**	417.6
	20NG-Vehicle	451.0	447.8	**460.5(s)**	446.7
	Reuters-1804	484.9	487.4	**488.7(s)**	482.0
	RCV1-2000	*473.0*	*467.6*	**476.9(s)**	473.2
	Spam-1000	487.0	*481.4*	**487.5(s)**	483.5
k-NN Consumer	**20NG-WinXwin**	*377.6*	**397.9(s)**	*389.5*	393.5
	20NG-Comp	*357.7*	355.1	*360.5*	**365.5(f)**
	20NG-Talk	*360.8*	376.8	**379.9(f)**	373.1
	20NG-Vehicle	*382.4*	403.0	*398.4*	**403.2(f)**
	Reuters-1804	*473.8*	**481.9(s)**	*471.8*	476.1
	RCV1-2000	*449.5*	**465.5(s)**	461.0	459.1
	Spam-1000	**462.4(f)**	462.0	462.0	454.0
Average Rank		**2.24**	**2.83**	**2.02**	**2.90**

Cases in which an actual active learning selection strategy is outperformed by random selection are highlighted using italics.

The results shown in Table 2 reinforce what was apparent from Figures 2 to 8. There are a wide range of performance levels achieved when different combinations of selectors and consumers are used. This clearly answers Q1 given above and shows that the reusability problem definitely exists for text classification problems.

4.2. Identifying the Best Selector

We move now to answer questions Q3a: *if the consumer is known, which selector should be used?* and Q3b: *if the consumer is unknown, which selector should be used?* from the list above. The results in Table 2 and Figures 2 to 8, show that when the naïve Bayes and the SVM consumers are used, self-selection is better than both foreign-selection and random selection on all seven datasets. When the k-NN consumer is used, results are a little more mixed. On three out of the seven datasets, the best selector is the k-NN selector. On the 20NG-Comp and 20NG-Vehicle datasets random selection performs best; on the 20NG-Talk the SVM selector performs best; and on the Spam-1000 dataset the naïve Bayes selector performs best. One reason why self-selection may not be ideal when a k-NN classifier is used is that the k-NN classifier is a local learner and has less bias to the examples used to train it. For this reason it may not gain as much benefit as the SVM and naïve Bayes global learners from the self-selected labelled training examples.

Overall it can be concluded from Table 2 that the advantage of self-selection over foreign-selection is supported by the evidence that in 17 out of the 21 combinations of consumers and datasets, the best performance is achieved on the training set produced by the selector of the same type of classifier. All of this evidence answers Q3a: if if the consumer is known self-selection should be used, i.e. the same type of classifier that will ultimately be used in the text classification application should be used in the active learning selection strategy.

Considering Q3b (*if the consumer is unknown, which selector should be used?*) requires a little more analysis. The performance of the four selectors were ranked for each consumer-dataset pair and average ranks were calculated across all 21 consumer-dataset pairs. These are shown in the final row of Table 2. The average ranks for the naïve Bayes selector, k-NN selector, SVM selector and random selector are 2.24, 2.83, 2.02 and 2.90 respectively. The results suggest an answer to Q3a: given no other information about how the labelled dataset arising from active learning is to be used, an SVM selector should be used.

It is also interesting to examine the difference in performance of classifier-based selectors and random selection. We can see from Table 2 that classifier-based foreign-selection is worse than random selection in a number of cases, particularly when the k-NN classifier is used as either the consumer or selector. This may be attributed to the fact that both the SVM and the naïve Bayes classifiers are eager-learners, while the k-NN classifier is a lazy-learner:

Table 3: Percentage overlap of the labelled sets between different selectors.

Dataset	$\mathbf{OL(S_{NB}, S_{SVM})}$	$\mathbf{OL(S_{NB}, S_{k-NN})}$	$\mathbf{OL(S_{SVM}, S_{k-NN})}$
20NG-WinXwin	54.2%	43.0%	41.3%
20NG-Comp	47.6%	36.2%	35.3%
20NG-Talk	61.8%	51.7%	50.3%
20NG-Vehicle	54.9%	44.5%	42.0%
Reuters-1804	67.2%	65.4%	64.5%
RCV1-2000	62.1%	54.6%	57.1%
Spam-1000	65.1%	60.8%	69.8%

an informative example for an eager-learner may not be an informative example for a lazy-learner. In order to explore this possibility further the overlap between the 500 examples selected by the SVM selector, the naïve Bayes selector and the k-NN selector was calculated for each of the seven datasets used in our experiments (overlap is calculated as the Jaccard index (Jaccard, 1912; Kelleher et al., 2015) between the selections, i.e. the number of examples that are present in both selections as a percentage of the total number of unique examples across both selections). Table 3 shows the overlap of L_{S_i} selected by the selector S_i and L_{S_j} selected by the selector S_j, denoted by $OL(S_i, S_j)$. Table 3 shows that there is a considerable difference in the labelled datasets that arise from different selectors. For all but one dataset, the overlap of examples selected by the SVM selector and the naïve Bayes selector are higher than both the overlap of examples selected by the SVM selector and the k-NN selector and the overlap of examples selected by the naïve Bayes selector and the k-NN selector. This suggests that an eager-learner and a lazy-learner have different preferences for informative examples and explains the poor performance when the two types of classifiers are mixed.

4.3. Identifying the Best Consumer

Our next consideration is which is the best consumer to use with datasets generated using active learning, i.e. to answer Q4a (*if the selector is known, which consumer should be used?*) and Q4b (*if the selector is unknown, which consumer should be used?*) given previously. To answer these questions we reuse the experimental results presented in Section 4.2, but in Table 4 present them in a slightly different way. Here we use *self-reuse* to refer to scenarios in which the consumer classifier is of the same type as the selector classifier

and *foreign-reuse* to refer to scenarios in which the classifiers are different. If a classifier achieves the best performance on a labelled training set produced by active learning and uses the same classifier type as the selector in the active learning process then we say that self-reuse is the best sample reuse scenario. Otherwise, if the best performance achieved from a labelled set using a consumer classifier different from that used by the active learning process which selected them, we say that foreign-reuse is the best sample reuse scenario.

In Table 4 the best consumer for each selector-dataset pair is shown in bold where a parenthesised (s) or (f) refer to self-reuse and foreign-reuse respectively (when a random selector is used all reuse is considered foreign-reuse).

The results show that self-reuse is not always the best reuse scenario and these results are consistent with the finding in Tomanek & Morik (2010). Self-reuse is better than foreign-reuse for only 8 out of the 21 selector-dataset pairs, while foreign-reuse is better than self-reuse for the remaining 13 selector-dataset pairs. This effectively answers Q2 above indicating that homogeneous systems (in which the consumer and selector classifiers are the same) are not always the best solution in active learning scenarios. This result also provides the answer to question Q4a: when the type of the selector is known, the best consumer to use is not necessarily the same type of classifier as used in active learning selection.

It is important to remember the distinction between self-selection and self-reuse here, as this result can at first seem somewhat contradictory to the answer to Q3a presented in Section 4.2 which recommended self-selection. In answering Q3a we considered the scenario where the consumer was fixed and we wished to determine the best selector to use, while in considering Q4a we consider the opposite scenario where the selector is fixed and we wish to determine which is the best consumer to use. One plausible reason for the recommendation against self-reuse is that the best classifier for one task not only depends on the training data used, but also on the power of the classifier itself. In these experiments foreign-reuse almost always refers to the use of a naïve Bayes classifier which is not all-together surprising as there is considerable evidence that naïve Bayes classifiers are particularly well suited to the kind of text classification problems we consider in this work.

The last row of Table 4 shows the average rank of the three consumers across all of the 28 selector-dataset combinations, including the random selector. It can be seen that overall the best consumer is the naïve Bayes

Table 4: Results for identifying the best **consumer**. The best consumer for each selector-dataset pair is shown in bold where a parenthesised (s) or (f) refer to self-reuse and foreign-reuse respectively (when a random selector is used all reuse is considered foreign-reuse).

		NB Consumer	**SVM Consumer**	**k-NN Consumer**
NB Selector	**20NG-WinXwin**	**458.5(s)**	416.6	377.6
	20NG-Comp	**429.3(s)**	407.8	357.7
	20NG-Talk	**447.3(s)**	427.2	360.8
	20NG-Vehicle	**470.3(s)**	451.0	382.4
	Reuters-1804	**488.8(s)**	484.9	473.8
	RCV1-2000	**483.8(s)**	473.0	449.5
	Spam-1000	481.5	**487.0(f)**	462.4
k-NN Selector	**20NG-WinXwin**	**441.4(f)**	409.9	397.9
	20NG-Comp	**413.6(f)**	397.1	355.1
	20NG-Talk	**430.6(f)**	417.06	376.8
	20NG-Vehicle	**459.2(f)**	447.8	403.0
	Reuters-1804	**488.3(f)**	487.4	481.9
	RCV1-2000	**480.1(f)**	467.6	465.5
	Spam-1000	472.0	**481.4(f)**	462.0
SVM Selector	**20NG-WinXwin**	**451.3(f)**	437.9	389.5
	20NG-Comp	**426.6(f)**	413.1	360.5
	20NG-Talk	**439.5(f)**	434.0	379.9
	20NG-Vehicle	**467.1(f)**	460.5	398.4
	Reuters-1804	484.2	**488.7 (s)**	471.8
	RCV1-2000	**479.4(f)**	476.9	461.0
	Spam-1000	458.6	**487.5(s)**	462.0
RS Selector	**20NG-WinXwin**	**443.7(f)**	423.8	393.5
	20NG-Comp	**421.9(f)**	402.0	365.5
	20NG-Talk	**425.4(f)**	417.6	373.1
	20NG-Vehicle	**461.2(f)**	446.7	403.2
	Reuters-1804	480.5	**482.0(f)**	476.1
	RCV1-2000	**477.3(f)**	473.2	459.1
	Spam-1000	460.8	**483.5(f)**	454.0
Average Rank		**1.25**	**1.79**	**2.96**

Table 5: An examination of the performance of selector-consumer pairs on the seven datasets used in this study. The best performance on each dataset is highlighted in bold.

	NB Consumer				SVM Consumer				k-NN Consumer			
	NB	kNN	SVM	RS	NB	kNN	SVM	RS	NB	kNN	SVM	RS
20NG-WinXwin	**459**	441	451	444	417	410	438	424	378	398	390	394
20NG-Comp	**430**	414	427	422	408	397	413	402	358	355	361	366
20NG-Talk	**447**	431	440	425	427	417	434	418	361	377	380	373
20NG-Vehicle	**470**	459	467	461	451	448	461	447	382	403	398	403
Reuters-1804	**489**	488	484	481	485	487	489	482	474	482	472	476
RCV1-2000	**484**	480	480	477	473	468	477	473	450	466	461	460
Spam-1000	482	472	459	461	487	481	**488**	484	462	462	462	454

consumer, followed by the SVM consumer. This result provides strong evidence for an answer to Q4b that if the type of the selector is unknown, a naïve Bayes classifier is the best choice to use as the consumer. This result flirts dangerously with the thorny question of what is the best classifier to use for text classification problems, which is not a question that we can ever hope to answer definitively. Given the caveat that the result only applies to the, albeit reasonably varied, set of classification problems we used in this study, however, it remains a useful result.

While it is not one of the core questions explored in this work, given the range of experiments performed it is worth considering which selector-consumer pair performs best across all of the datasets used. Table 5 recasts the data presented in Table 2 and Table 4 to address this question. The performance of the best selector-consumer pair is highlighted in bold for each dataset. It can be seen that on six out of the seven datasets considered in this work the best performing combination is a naïve Bayes selector paired with a naïve Bayes consumer. On the Spam-1000 dataset, the best selector and consumer pair is the SVM selector with the SVM consumer. One possible reason for this might be that the Spam-1000 dataset contains emails which have significantly different characteristics to more general text samples (e.g. the length of the documents and the vocabulary used).

4.4. Efficiency of Selectors

As well as considering the performance of active learning selector-consumer pairs it is also useful to examine the efficiency of different selectors, as timely response times are important to make active learning practical in real appli-

cation scenarios – users playing the role of the oracle will not wait indefinitely to be asked to label examples. To this end experiments were conducted to compare the execution time for different selectors when selecting 500 examples for labelling on each of the seven datasets described in Section 3.2. Experiments were conducted on a Macbook running Mac OS X 10.6.6, using an Intel Xeon CPU i3@3.06GHz with 4.0 GB 1333 MHz DDR3 RAM. In each experiment, for each dataset, all selectors were allowed to select 500 examples for labelling. Each experiment was repeated three times and average execution times (in seconds) are reported for each dataset in Table 6.

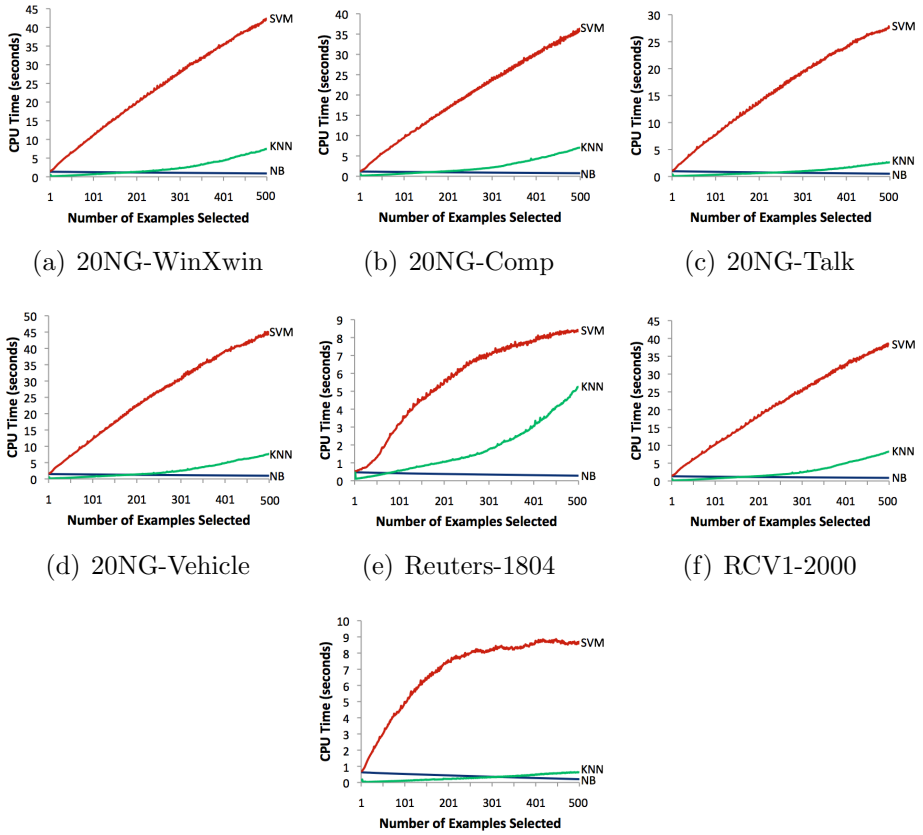
Table 6: Time performance (in seconds) on the seven datasets. The best performance on each dataset is highlighted in bold.

Dataset	**NB Selector**	**SVM Selector**	**k-NN Selector**
20NG-WinXwin	**553**	11,659	1,235
20NG-Comp	**468**	9,895	1,172
20NG-Talk	**364**	7,988	488
20NG-Vehicle	**611**	12,805	1,322
Reuters-1804	**183**	2,806	892
RCV1-2000	**536**	10,683	1,341
Spam-1000	201	3,429	**148**

From Table 6 it is apparent that, among the selectors compared, the naïve Bayes selector is the most efficient, mainly due to the fact that the training and classification processes of the naïve Bayes classifier are particularly efficient. The k-NN selector is the second most efficient selector as the repeated retraining required in active learning is of especially low intensity – new examples are simply added to the labelled set. The SVM selector is the least efficient selector with a significantly longer execution time as, in every active learning iteration, the SVM model needs to be retrained and the training time increases with the number of examples in the training set.

Graphs of the execution time taken for each active learning iteration for each selector on each dataset are shown in Figure 9. These graphs show that the execution time of the SVM and k-NN selectors increasing as the size of the labelled set grows, while the execution time for the naïve Bayes selector remains relatively stable. These results suggest the use of a naïve Bayes selector in applications in which response time is of particular importance. Overall, however, the maximum waiting time between labelling batches for

these datasets (when an SVM is used for the 20NG-Vehicle dataset) is still less than a minute which is probably reasonable.



(a) 20NG-WinXwin (b) 20NG-Comp (c) 20NG-Talk

(d) 20NG-Vehicle (e) Reuters-1804 (f) RCV1-2000

(g) Spam-1000

Figure 9: CPU time of three selectors on seven datasets. Axes are zoomed for resolution.

5. Conclusions and Future Work

Understanding the reusability of training examples generated using an active learning selection strategy that uses a specific classifier is an interesting problem that is important in practical machine learning applications. We have empirically studied the performance of different selector-consumer classifier pairs on a range of text classification problems. Based on these experiments the following conclusions can be drawn:

- When using active learning to generate training examples for a particular consumer classifier, it is best to use the same type of classifier in the selection process.

- Overall, with a view to good reusability by different classification algorithms, the best classifier to be used in an active learning selection strategy is the SVM classifier.

- Local and global classifiers don't mix well with regard to reusability. Occasionally, the performance of the k-NN classifier trained on a set of examples selected by active learning using a different type of classifier (for example, an SVM classifier or a naïve Bayes classifier) can be worse than the performance of the k-NN classifier trained on randomly selected examples.

- When reusing examples selected using active learning, it is not necessary to use the same type of classifier as in the selection process. The use of the same classifier in reuse as in selection does not guarantee the best performance.

- Overall, the best classifier to train with examples selected by active learning is the naïve Bayes classifier which can work well with different sets of training examples produced by active learning methods using different classifiers in selection. This indicates that the naïve Bayes classifier has less dependency on the training data and is less sensitive to the data used to train it compared to the SVM classifier and the k-NN classifier.

- The best classifier pair in active learning selection and sample reuse is an naïve Bayes classifier in active learning selection with the same naïve Bayes classifier in sample reuse.

In summary, if information about the classifier used in active learning for generating the labelled training examples is unknown, a naïve Bayes classifier is a good choice as the consumer classifier. If we want to use active learning to build a training set and the classifier which will be trained on the labelled set has been decided, then the same type of classifier should be used in active learning selection. If the classifier to be trained on the examples selected using active learning is unknown, an SVM-based active learning selection strategy is the best choice for constructing the labelled training set in order

to ensure maximum reusability. If efficiency is of high concern then a naïve Bayes classifier may be a better choice as the basic classifier used in active learning selection.

There are a number of directions that we intend to explore in the future. First, previous work has shown that both model relatedness and sample similarity cannot explain reusability. It would be interesting to discover the supporting factors for reusability, i.e., what factors contribute to higher reusability. Second, the sample selection bias induced by basic classifiers used in active learning is the main reason for poor reusability. Ensemble methods, or classifier-free selection strategies, might be helpful in solving this problem. Third, it would be interesting to examine the reusability on multi-class and multi-label classification. Fourth, the work presented in this article has focused exclusively on text classification problems. While the reusability problem remains an issue for other types of classification problems, the specific findings in this article in relation to the best selector and consumer classifiers may not hold for other types of classification problems. The nature of text classification problems leads to high–dimensional, sparse feature vectors, which favour particular classification algorithms (for example naïve Bayes and SVM) over others. Therefore it would be very interesting to repeat the experiments performed in this article for other types of classification problems. Finally, because the use of a classification model at the heart of uncertainty sampling causes the reusability problem in the first place, it would be interesting to investigate the impact of selection strategies that do not rely on a classification model (such as those described by Hu et al. (2010a) and Lughofer (2012b)) on the performance of consumer models.

References

Ando, R. K., & Zhang, T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, *6*, 1817–1853.

Andrew, I., Schein, & Ungar, L. (2005). Active learning for multi-class logistic regression.

Angluin, D. (1988). Queries and concept learning. *Machine Learning*, *2*, 319–342.

Baldridge, J., & Osborne, M. (2004). Active learning and the total cost of annotation. In *Proceedings of EMNLP 2004* (pp. 9–16).

Baram, Y., El-Yaniv, R., & Luz, K. (2004). Online choice of active learning algorithms. *J. Mach. Learn. Res.*, *5*, 255–291.

Campbell, C., Cristianini, N., & Smola, A. J. (2000). Query learning with large margin classifiers. In *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 111–118). Morgan Kaufmann Publishers Inc.

Cebron, N., & Berthold, M. (2008). Active learning for object classification: from exploration to exploitation. *Data Mining and Knowledge Discovery*, .

Cebron, N., & Berthold, M. R. (2006). Adaptive active classification of cell assay images. *Lecture notes in computer science ISSN 0302-9743*, .

Chu, W., Zinkevich, M., Li, L., Thomas, A., & Tseng, B. (2011). Unbiased online active learning in data streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 195–203). ACM.

Cohn, D., Atlas, L., & Ladner, R. (1994). Improving generalization with active learning. *Machine learning*, *15*, 201–221.

Cohn, D., Ghahramani, Z., & Jordan, M. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, *4*, 129–145.

Dasgupta, S., & Hsu, D. (2008). Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*.

Delany, S. J., Cunningham, P., Tsymbal, A., & Coyle, L. (2005). A case-based technique for tracking concept drift in spam filtering. (pp. 3–16). Springer London.

Donmez, P., Carbonell, J., & Bennett, P. (2007). Dual strategy active learning. (pp. 116–127).

Drucker, H., Wu, D., & Vapnik, V. (1999). Support vector machines for spam categorization. *Neural Networks, IEEE Transactions on*, *10*, 1048–1054.

Ertekin, S., Huang, J., & Giles, C. L. (2007). Active learning for class imbalance problem. In *Proc. of the 30th International Conference on Research and Development in Information Retrieval (ACM SIGIR 2007)*. Amsterdam, Netherlands.

Freund, Y., Seung, H., Shamir, E., & Tishby, N. (1997). Selective sampling using the query by committee, .

Guyon, I., Cawley, G., Dror, G., & Lemaire, V. (2010). Design and analysis of the WCCI 2010 active learning challenge. In *Neural Networks (IJCNN), The 2010 International Joint Conference on* (pp. 1–8).

Hoi, S. C., Jin, R., & Lyu, M. R. (2009). Batch mode active learning with applications to text categorization and image retrieval. *IEEE Transactions on Knowledge and Data Engineering*, *21*, 1233–1248.

H.S.Seung, M.Opper, & H.Sompolinsky (1992). Query by committee. In *In Proceedings of the Fifth Workshop on Computational Learning Theory* (pp. 287–294). San Mateo, CA: Morgan Kaufmann.

Hu, R., Delany, S. J., & Mac Namee, B. (2010a). EGAL: exploration guided active learning for TCBR. In *Case-Based Reasoning. Research and Development* (pp. 156–170). Springer Berlin / Heidelberg volume 6176 of *Lecture Notes in Computer Science*.

Hu, R., Mac Namee, B., & Delany, S. J. (2008). Sweetening the dataset: Using active learning to label unlabelled datasets. In *Proceedings of the 19th Irish Conference on Artificial Intelligence and Cognitive Science (AICS '08)*.

Hu, R., Mac Namee, B., & Delany, S. J. (2010b). Off to a good start: Using clustering to select the initial training set in active learning. In *Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference (FLAIRS 2010)* (pp. 26–31). AAAI.

Hu, W., Hu, W., Xie, N., & Maybank, S. (2009). Unsupervised active learning based on hierarchical graph-theoretic clustering. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *39*, 1147–1161.

Jaccard, P. (1912). The distribution of the flora in the alpine zone. *New phytologist*, *11*, 37–50.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In I. Bratko, & S. Dzeroski (Eds.), *Proceedings of ICML-99, 16th International Conference on Machine Learning* (pp. 200–209). Bled, SL: Morgan Kaufmann Publishers, San Francisco, US.

Kang, J., Ryu, K., & Kwon, H. (2004). Advances in knowledge discovery and data mining. chapter Using Cluster-Based Sampling to Select Initial Training Set for Active Learning in Text Classification. (pp. 384–388). Springer volume 3056.

Kelleher, J. D., Mac Namee, B., & D'Arcy, A. (2015). *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*. MIT Press.

Laws, F., & Schütze, H. (2008). Stopping criteria for active learning of named entity recognition. In *Proceedings of Coling*.

Lewis, D., & Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In *In Proceedings of the Eleventh International Conference on Machine Learning* (pp. 156, 148).

Lewis, D. D., & Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *Proc 17th annual International ACM SIGIR conference on Research and Development in Information Retrieval* (pp. 3–12). Springer-Verlag NY.

Lughofer, E. (2012a). Hybrid active learning for reducing the annotation effort of operators in classification systems. *Pattern Recognition*, *45*, 884 – 896.

Lughofer, E. (2012b). Single-pass active learning with conflict and ignorance. *Evolving Systems*, *3*, 251–271.

Nguyen, H. T., & Smeulders, A. (2004). Active learning using pre-clustering. *In Proceedings of the 21st International Conference on Machine Learning*, (pp. 623–630).

Nigam, K., & McCallum, A. (1998). Pool-based active learning for text classification, .

Novak, B., Mladenîć, D., & Grobelnik, M. (2006). Text classification with active learning. (pp. 398–405).

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP* (pp. 79–86).

Porter, M. F. (1980). An algorithm for suffix stripping. *Program: electronic library and information systems*, *14*, 130–137.

Probst, & Ghani (2007). Towards 'interactive' active learning in multi-view feature sets for information extraction. (pp. 683–690).

Raghavan, H., Madani, O., & Jones, R. (2006). Active learning with feedback on both features and instances. *J. Mach. Learn. Res.*, *7*, 1655–1686.

Rose, T., Stevenson, M., & Whitehead, M. (2002). The reuters corpus volume 1 - from yesterday's news to tomorrow's language resources. *IN PROCEEDINGS OF THE THIRD INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION*, (pp. 29—31).

Roy, N., & McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. In *Proc. 18th International Conf. on Machine Learning* (pp. 441–448). Morgan Kaufmann, San Francisco, CA.

Schneider, J. (2009). Active learning for fitting simulations to observational data. In *IJCAI workshop on Machine Learning and AI Applications in Astrophysics and Cosmology*. Pasadena, CA.

Schohn, G., & Cohn, D. (2000). Less is more: Active learning with support vector machines. In *Proc. 17th International Conf. on Machine Learning* (pp. 839–846). Morgan Kaufmann, San Francisco, CA.

Schütze, H., Velipasaoglu, E., & Pedersen, J. O. (2006). Performance thresholding in practical text classification. In *Proceedings of the 15th ACM international conference on Information and knowledge management* (pp. 662–671). Arlington, Virginia, USA: ACM.

Segal, R., Markowitz, T., & Arnold, W. (2006). Fast uncertainty sampling for labeling large e-mail corpora. In *CEAS 2006: Conference on Email and Anti-Spam*.

Settles, B. (2009). *Active Learning Literature Survey*. Technical Report University of Wisconsin–Madison.

Tomanek, K. (2010). *Resource-aware annotation through active learning*. Text Technical University of Dortmund.

Tomanek, K., & Morik, K. (2010). Inspecting sample reusability for active learning. In *JMLR: Workshop and Conference Proceedings 10 (2010) Workshop on Active Learning and Experimental Design* (pp. 1–12).

Tomanek, K., & Olsson, F. (2009). A web survey on the use of active learning to support annotation of text data. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing* (pp. 45–48). Boulder, Colorado: Association for Computational Linguistics.

Tomanek, K., Wermter, J., & Hahn, U. (2007). An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data, . *3*.

Tong, S., & Chang, E. (2001). Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international conference on Multimedia* (pp. 107–118). Ottawa, Canada: ACM.

Tong, S., & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, *2*, 45–66.

Vlachos, A. (2008). A stopping criterion for active learning. *Comput. Speech Lang.*, *22*, 295–312.

Voorhees, E. M. (1986). *The effectiveness and efficiency of agglomerative hierarchic clustering in document retrieval*. Ph.D. thesis Cornell University.

Warmuth, M., Liao, J., Ratsch, G., Mathieson, M., Putta, S., & Lemmen, C. (2003). Active learning with support vector machines in the drug discovery process. *Journal of Chemical Information and Computer Sciences*, *43*, 667–673.

Xu, Z., Akella, R., & Zhang, Y. (2007). Incorporating diversity and density in active learning for relevance feedback. (pp. 246–257).

Xu, Z., Yu, K., Tresp, V., Xu, X., & Wang, J. (2003). Representative sampling for text classification using support vector machines. In *Advances in Information Retrieval* (p. 11). Springer.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval*, *1*, 69–90.

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '99* (pp. 42–49). Berkley.

Yu, K., Zhu, S., Xu, W., & Gong, Y. (2008). Non-greedy active learning for text categorization using convex ansductive experimental design. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 642, 635). Singapore, Singapore: ACM.

Zhu, J., Wang, H., Hovy, E., & Ma, M. (2010). Confidence-based stopping criteria for active learning for data annotation. *ACM Transactions on Speech and Language Processing*, *6*, 1–24.

Zhu, J., Wang, H., & Tsou, B. (2008). Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In *Proceedings of the 22nd International Conference on Computational Linguistics* (pp. 1137–1144).