2011-10

Active Learning for Text Classification

Rong Hu
Technological University Dublin

Recommended Citation

Hu, R. (2011). *Active Learning for Text Classification. Doctoral Thesis. Technological University Dublin. doi:10.21427/D70K5Z*

Active Learning for Text Classification

by

Rong Hu

Supervisors: Dr. Brian Mac Namee
Dr. Sarah Jane Delany
Prof. Pádraig Cunningham

School of Computing

Dublin Institute of Technology

A thesis submitted for the degree of

Doctor of Philosophy

October, 2011

I would like to dedicate this thesis to my loving families ...

Declaration

I certify that this thesis which I now submit for examination for the award of Doctor of Philosophy, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work.

This thesis was prepared according to the regulations for postgraduate study by research of the Dublin Institute of Technology and has not been submitted in whole or in part for an award in any other Institute or University.

The work reported on in this thesis conforms to the principles and requirements of the institute's guidelines for ethics in research.

The Institute has permission to keep, to lend or to copy this thesis in whole or in part, on condition that any such use of the material of the thesis be duly acknowledged.

Signature____________________________ Date____________________________

Acknowledgements

This thesis would not exist without the help, friendship and support of many people. I am happy to take this opportunity to thank those who have influenced and guided me during my graduate career.

First and foremost I would like to acknowledge my supervisors – Dr. Brian Mac Namee, Dr. Sarah Jane Delany and Prof. Pádraig Cunningham who have had the greatest impact on my academic development during my time at Dublin Institute of Technology (DIT). I want to express my gratitude to Prof. Pádraig Cunningham for the invaluable advice and insight he provided throughout the research. I am extraordinarily grateful to my primary supervisors: Dr. Brian Mac Namee and Dr. Sarah Jane Delany. I thank they for believing in me and giving me the great chance to pursue my Ph.D. in DIT. They are open-minded, supportive, resourceful, and caring. They had been tremendous supervisors and collaborators, providing me with invaluable guidance and advice about research, teaching and academic skills in general. I appreciated their constant support, their brilliant ideas, their deep knowledge and understanding of machine learning and text classification. They are tireless and enthusiastic in their position as supervisors and have always been available. They always replied very quickly and very efficiently to the

numerous emails I sent to them during the last four years. They always read my paper drafts thoroughly, giving constantly invaluable advice and comments in content, structure, and style. They gave me the confidence and knowledge I needed to turn my dreams into reality. I feel unbelievably lucky to have been their student and I owe them a great many heartfelt thanks.

The Applied Intelligence Research Centre (AIRC, originally the DIT AI Group) is a wonderful place to work and my colleagues gave me all the support I needed. I would like to thank the past and present members of AIRC that I have had the great fortune of knowing: Mark Dunne, Kenneth (Ken) Kennedy, Yan Li, Patrick (Paddy) Lindstrom, Robert Ross, Niels Schütte, Gurpreet Singh, Colm Sloan and Alexey Tarasov. They have been great to knock around ideas with, to learn from, as well as being good friends. And it has been a pleasure to have them around with their contagious positive attitude. My special thanks to Paddy. I am deeply indebted to him for his support over the years and last for the thesis. We have discussed many interesting topics, and explored conferences together. I am particularly grateful to Ken for the indispensable advice he provided, and for his diligence in proof-reading several drafts of my papers. Ken has been showing me his generosity, humor and skills. We had many interesting conversations over the years. I would also like to thank pc-turing and pc-cavan for their hard work and 24/7 dedication, to run experiments.

I am also very grateful to people who have indirectly influenced my research. I would like to thank all the staffs in School of Computing for their help and support and Denise Murray deserves special thanks for making everything easier. I would like to thank Dr. John Kelleher for all that I learned in his artificial intelligence class. I thank Aoife D'Arcy and Joe Condon for generously sharing their expertise and their lucid view of the areas of probability and statistics. I am very grateful to Dr. Mary McNamara, the Head of Graduate Research for organizing all the wonderful research events. I thank Graduate Research School Office especially Denise Farrell-Smith and Raffaella Salvante for being always ready to help and knowing how to.

Outside of DIT, my experience of the 13th Machine Learning Summer School (MLSS) at Churchill College in the University of Cambridge was both enlightening and rewarding. It made such a big difference to me to actually meet so many of the big names in the field – it not only allowed me to get to know them personally, but also urged me to do better in my research career. I would like to particularly thank Tom Minka who not only gave us great talks on approximate inference but also encouraged me on the study of active learning.

During my 23 years in school, I was fortunate to encounter teachers that motivated me, shaped my thinking, and greatly influenced my future: Xiuzhen Li, Linghui Wang, Ronghuang Huang, Muzhi Li, Jianxin He, Jie Yu, Changjie Tang and Mr. Yun. Each of them deserves more thanks that I can ever express. I am very grateful to Xiuzhen Li who advised me to go to graduate school even when I was a little girl in the primary school. I give special thanks to Changjie Tang, my supervisor at Sichuan University in my graduate study for the master degree, who initiated me into research, encouraged me to pursue a doctoral degree, and helped me in many ways to accomplish my goals. I am extraordinarily grateful to Mr. Yun for showing me a bright way to pursue.

I am also pleased to have spent a portion of my time in ISPCC and AgeAction. I would like to deeply thank Lisa Collins, Amy Callahan and Margaret Fitzpatrick for their support and for showing me their confidence on my work.

Let me just say that I send my heartfelt gratitude to all my dear friends, as well as a sincere apology that I am not listing you all by name. I will just mention a few people who have been extremely supportive during my study in Ireland: Alan Flynn Aldo, Marzena Cybulska, Qiaohuang Cheng, Fuhu Deng, Jianhua Deng, Yi (熠) Ding, Yi(一) Ding, Ketao Gong, Xiaohua Guo, Xiang Han, Jingjing Huang, NengLian Jin, Bo Liu, Hongliang Liu, Xiang Liu, Xin Liu, Yupeng Liu, Xiao Meng, Jin Su, Chenqi Wang, Qi Wang, Huali Wu, Jianfeng Wu, Jianhua Xu, Wei Yan, Tiedong Yang, Lingjian Zeng, Erqiang Zhou and Qianling Zhou. I would like to thank Alan Flynn Aldo for helping my transition into a different culture and for all his kind caring. Fuhu Deng, Jianhua Deng, Yi (熠) Ding, Yi(一) Ding, Jianfeng Wu, Tiedong Yang and Erqiang Zhou, deserve my infinite gratitude for never, ever doubting me, for invaluable support, encouragement, and cheering-me-up whenever necessary. Every step of the way, they have made me laugh, and made me food. They made the house in Ireland a comfortable home away from home. The sweet memory of our improvised meals, and our laughs will never desert me.

Many friends-for-life, although dispersed all over the world, lightened my days with their lovely emails, phone calls, and online chats. I also owe a great deal of thanks to friends who have been a steady source of inspiration and support for me: Qi Chen, Sheng Chen, Lili Cui, Jiaxiang Du, Xin Gu, Wenjing Luo, Qianghai Ren, Huimin Xu, Mingyu Xu, Huabin Yan, Li Zhang, Hui Zeng and too many others – you know who you are!

I owe a special debt of gratitude to my parents for their enduring love and support. Completing a Ph.D. is like running a marathon. Without their sacrifices and inspiration, it would never have been possible for me to reach this point. They always support me in whatever decisions I made, even when the decisions took me in a different direction than they had originally intended. By the same token, I am greatly indebted to my parents-in-law for their love and support throughout my career as a Ph.D. research student. I also owe a great deal of thanks to my brothers and sisters-in-law: Hua Hu, Baohua Zhou, Da Hu and Ronghua Zhang. I would not have had the strength to complete this work without the love and support of my family. I am forever grateful of their belief in

me.

My deepest gratitude and appreciation is reserved for my husband Youli Song and my son Zeshui Song, for the infinite reserves of love, support, and encouragement. Without their love, support, and sacrifice, this work would have never come to fruition. I would like to thank Youli for sharing all these years and for brightening my life. I am eternally grateful to Zeshui for being a good boy far away parents and always cheering me with sweet smiles and voices. I am deeply affected by their trust and love. I dedicate this thesis to them.

Finally, I would like to thank the funding agency that made this research possible: Science Foundation Ireland under Grant No.
07/RFP/CMSF718.

Abstract

Text classification approaches are used extensively to solve real-world challenges. The success or failure of text classification systems hangs on the datasets used to train them, without a good dataset it is impossible to build a quality system. This thesis examines the applicability of active learning in text classification for the rapid and economical creation of labelled training data. Four main contributions are made in this thesis.

First, we present two novel selection strategies to choose the most informative examples for manually labelling. One is an approach using an advanced aggregated confidence measurement instead of the direct output of classifiers to measure the confidence of the prediction and choose the examples with least confidence for querying. The other is a simple but effective exploration guided active learning selection strategy which uses only the notions of density and diversity, based on similarity, in its selection strategy.

Second, we propose new methods of using deterministic clustering algorithms to help bootstrap the active learning process. We first illustrate the problems of using non-deterministic clustering for selecting initial training sets, showing how non-deterministic clustering methods

can result in inconsistent behaviour in the active learning process. We then compare various deterministic clustering techniques and commonly used non-deterministic ones, and show that deterministic clustering algorithms are as good as non-deterministic clustering algorithms at selecting initial training examples for the active learning process. More importantly, we show that the use of deterministic approaches stabilises the active learning process.

Our third direction is in the area of visualising the active learning process. We demonstrate the use of an existing visualisation technique in understanding active learning selection strategies to show that a better understanding of selection strategies can be achieved with the help of visualisation techniques.

Finally, to evaluate the practicality and usefulness of active learning as a general dataset labelling methodology, it is desirable that actively labelled dataset can be reused more widely instead of being only limited to some particular classifier. We compare the reusability of popular active learning methods for text classification and identify the best classifiers to use in active learning for text classification.

This thesis is concerned using active learning methods to label large unlabelled textual datasets. Our domain of interest is text classification, but most of the methods proposed are quite general and so are applicable to other domains having large collections of data with high dimensionality.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Machine learning is an offshoot of *Artificial Intelligence* (AI)(McCarthy *et al.*, 1955). The primary goal of machine learning is to develop general purpose algorithms of practical value from a limited amount of data. The distinguishing factor in machine learning approaches to general AI is the use of data and the discovery of patterns in data. There are many examples of machine learning applications, such as medical diagnosis, fraud detection and weather prediction. Supervised learning and unsupervised learning are the two main areas of machine learning. Supervised learning is the machine learning task of generating a mapping from supervised or labelled training data to an output of classes or predictions. One core area of supervised learning is the classification task. The goal in classification is to create a function from input objects to output values, referred to as *labels* or *classes*. The mapping or the function is referred to as a *classifier* or a *model*. The input objects are things that are to be classified, also known as *instances*, *tuples* or *examples*. The machine learning approach to classification is to gather a set of examples with their classes known by manually labelling some number of examples. The set of labelled examples is known as a *training set*. Next a classifier is used together with the training set to generate a mapping from examples to labels. Then the trained classifier can be used to *classify* or *label* new, unseen examples.

Increasingly examples are available in free text form, the classification problem of determining the predefined categories of natural language documents, namely *text classification*, becomes more and more important. A classic example of text classification is categorizing news articles into topics such as politics and sports. However, the competence of a classification system relies significantly on the quality of the training data used. Building a training set requires a large number of historical labelled examples. Gathering such labelled collections can be time consuming and expensive. In many cases limited resources are available for collecting such data which makes it difficult to create classifiers for known problems. Instead of randomly picking examples to be manually labelled for building the training set, we have the option of carefully choosing examples from the pool that are to be labelled using active learning. In this work, we explore active learning methods to label textual datasets.

1.1 Reducing Labelling Costs in Text Classification

There are many situations where unlabelled examples are plentiful and cheap, but it is expensive and time-consuming to label these examples. For instance, it is easy to grab billions of webpages at essentially no cost, but much more costly to pay human annotators to label those documents with their topic categories. Likewise, it is simple to collect video clips data, but much harder to obtain good semantic content labels. It is also easier to obtain a large collection of compounds that may be good to cure some disease, but much more costly to run expensive biochemical tests to know which one really works. Fortunately, creating labelled datasets can be addressed using active learning. Active learning is a machine learning technique that can be used to build accurate classifiers or to label unlabelled datasets by selecting the most informative examples and querying their labels from human experts (the oracle). The application task we focus on is text classification.

A typical framework of active learning is shown in Figure 1.1. In this framework, the active learner starts from a small labelled dataset and with access to a large pool of unlabelled data and selects the most informative unlabelled data from the pool. Then the active learner can query the true label of the selected most informative examples from the oracle, remove those selected examples from the pool and add them into the labelled dataset. This process is repeated until some stopping criterion has been met. This is in contrast with the traditional method of labelling randomly selected material, namely *random sampling*, which are often referred to as *passive*

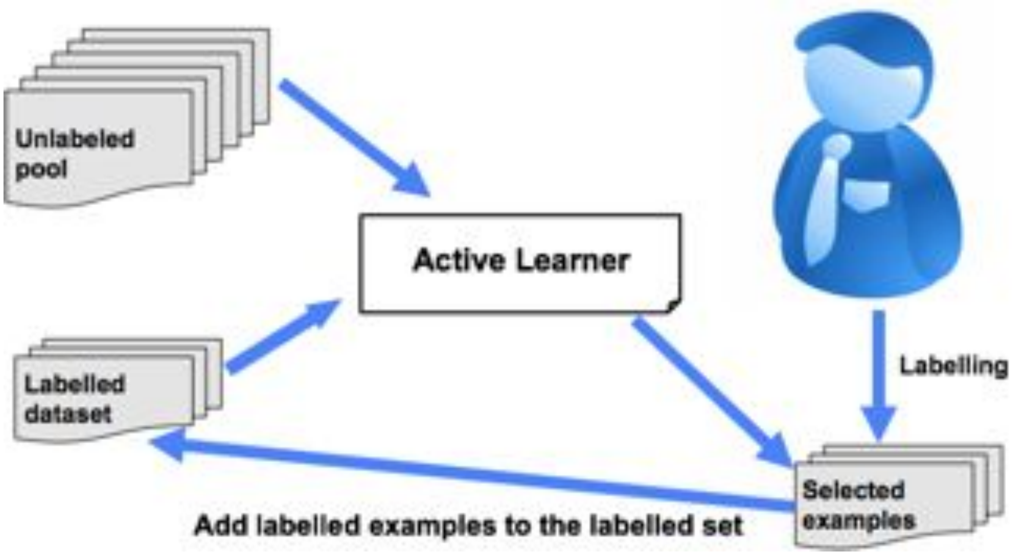


Figure 1.1: Active learning framework.

learning.

The benefits of active learning for labelling can include a reduction in the amount of data needed to achieve learning, improved predictive accuracy, and reduced execution time. These factors are of particular importance in the area of real-life machine learning applications. This thesis focuses on active learning labelling – a process that can benefit learning algorithms to create high-quality labelled datasets with minimum manual labelling efforts. Active learning for real-world dataset labelling should meet the following requirements (Tomanek, 2010):

- **Accuracy**: it should output a labelled dataset with high accuracy;
- **Efficiency**: it should have a reasonable interactive process with the oracle, which means fast selection cycles are preferred;
- **Reusability**: it should supply a labelled dataset which can be reasonably used by different applications.

We introduce active learning methods to select fewer good quality examples for human labelling than traditional methods which randomly select examples for labelling, thus alleviating the large amounts of labelling workload for human experts and reduce the scale of the datasets, and consequently reduce the computational cost of classifiers.

1.2 Contributions of the Thesis

In this work, we place a particular emphasis on understanding the implementation of active learning methods in text classification. The aim of this thesis is to explore various key aspects of active learning for the tasks of labelling large unlabelled textual datasets. The chapters that follow (i) formalise active learning and its applications to text classification, (ii) present the active learning scenarios and algorithms we have developed for labelling textual datasets, (iii) demonstrate the use of visualisation techniques in understanding active learning methods, (iv) investigate the reusability problem in active learning. This thesis claims that labelled dataset creation for supervised machine learning tasks can be accomplished using active learning. We explored three main hypotheses in this thesis:

- Active learning strategies that consider confidence measures can perform better than strategies that rely on the direct output of the classifier.
- Active learning strategies that focus on exploration can perform better than those that focus on exploitation in the early learning stage.
- The active learning process can be bootstrapped using deterministic clustering techniques.

We proposed different ways to explore these issues, developing new selection strategies and new methods to seed active learning process. Our main contributions on these initial hypotheses can be summarised in the following way:

- **Active learning literature review (Chapter 3)**: We presented a comprehensive review of active learning and how it is used in text classification.

- **Confidence-based selection strategy (Chapter 5)**: We developed a new selection strategy based on confidence measures (Hu *et al.*, 2009).

- **Exploration guided selection strategy (Chapter 6)**: We proposed a new selection strategy based on exploration that performs well in the early active learning stage where exploration is especially helpful (Hu *et al.*, 2010a).

- **Hybrid selection strategy (Section 9.2)**: We proposed a hybrid selection strategy combining an exploitation based selection strategy and an exploration based selection strategy which showed promising performance (Hu *et al.*, 2010b).

- **Use of visulisation techniques in active learning (Section 6.4)**: We demonstrated the usefulness of visulisation in understanding active learning selection strategies (Mac Namee *et al.*, 2010).

- **New method to seed active learning (Chapter 7)**: We proposed new methods of using deterministic clustering algorithms to help bootstrap the active learning process (Hu *et al.*, 2010c).

- **Understanding the reusability problem in active learning (Chap-

ter 8): We conducted a comprehensive analysis of the reusability performance of classifiers used in text classification when applied to active learning. Results showed that a labelled dataset from an active learning process can generally be used to build different types of classifiers than the specific classifier employed in the active learning selection strategy. It is best to use the same type classifier in the active learning selection as the one will be used in the sample reuse scenario.

- **Best classifiers in active learning (Chapter 8)**: We identified the best classifiers to use in active learning for text classification.

Finally, during this research, we have published our results at different conferences. The complete list is given in Section 1.4.

Although we place a particular emphasis on text classification applications, the techniques described throughout this thesis are quite general in nature. Many machine learning problems in natural language processing, bioinformatics and computer vision can be benefited from methods proposed in this work.

1.3 Outline of the Thesis

This thesis is structured as following:

- Chapter 2 provides a high level overview of concepts from supervised learning and text classification.
- Chapter 3 concisely surveys past research in the active learning domain. After a brief discussion of the general active learning process, an overview of common fields of active learning application is provided.
- Chapter 4 first describes the system framework design of ALL, our active-learning-based labelling system. Then it gives an initial evaluation of the proposed framework on a recipe dataset which shows that active learning methods can help in the creation of labelled datasets. Thirdly, a description of the experimental methods and datasets used in the empirical evaluations throughout this thesis is provided.
- Chapter 5 presents the design, implementation and evaluation of ACMS – a novel ensemble based active learning selection strategy using k-Nearest Neighbour classifier confidence measures and shows that ACMS compares favorably against current state-of-the-art methods.
- Chapter 6 first introduces EGAL, an exploration guided active learning selection strategy. Then it demonstrates the use of visualisation techniques in understanding active learning selection strategies.

- Chapter 7 addresses the question of how to select initial training sets for active learning. The two main objectives of Chapter 7 are to show that commonly used techniques are problematic and to show the superiority of deterministic clustering algorithms in creating initial training sets.

- Chapter 8 explores and discusses the reusability problem in active learning with the aim of identifying the best classifier in active learning in the context of text classification.

- Chapter 10 summarises key contributions of this work and highlights opportunities for additional research.

1.4 Publications

The thesis is supported by the following publications:

[**Hu *et al.* (2009)**] Hu, R., Delany, S.J., Mac Namee, B.: Sampling with confidence: Using k-NN confidence measures in active learning. In: Proceedings of the UKDS Workshop at the 8th International Conference on Case-based Reasoning (ICCBR 09). (2009) 181-192

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

[**Hu *et al.* (2010b)**] Hu, R., Lindstrom, P., Delany, S.J., Mac Namee, B.: Exploring the frontier of uncertainty space. At the AISTATS 2010 Workshop on Active Learning and Experimental Design (May 16, 2010; Sardinia, Italy).

[**Hu *et al.* (2010c)**] Hu, R., Mac Namee, B., Delany, S.J.: 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), AAAI. (2010) 26-31 (Best Student Paper Award)

Additional work related to the contribution is as following:

[**Zhang *et al.* (2008)**] Zhang, Q., Hu, R., Mac Namee, B., Delany, S.J.: Back to the future: Knowledge light case base cookery. In Schaaf, M., ed.: Proceedings of Workshop on Computer Cooking Contest, ECCBR'08. (2008) 239-248 (Champion of the 1st Computer Cooking Contest)

[**Hu *et al.* (2008)**] Hu, R., Mac Namee, B., Delany, S.J.: Sweetening the dataset: Using active learning to label unlabelled datasets. In Proceedings of the 19th Irish Conference on Artificial Intelligence and Cognitive Science. (2008) 53-62

[**Lindstrom *et al.* (2010)**] Lindstrom, P., Hu, R., Delany, S.J., Mac Namee, B.: SVM based active learning with exploration. At the AISTATS 2010 Workshop on Active Learning and Experimental Design (May 16, 2010; Sardinia, Italy).

[**Mac Namee *et al.* (2010)**] Mac Namee, B., Hu, R., Delany, S.J.: Inside the selection box: Visualising active learning selection strategies. At the NIPS 2010 Workshop on Challenges of Data Visualization (December 11, 2010; Whistler, BC).

As a summary, the contributions of this work, the corresponding chapters of this thesis and the publications are shown in Table 1.1.

Table 1.1: Contributions, corresponding chapters and publications.

Contribution	**Chapter**	**Publication**
Active learning literature review	Chapter 3	
Confidence-based selection strategy	Chapter 5	Hu *et al.* (2009)
Exploration guided selection strategy	Chapter 6	Hu *et al.* (2010a)
Hybrid selection strategy	Section 9.2	Hu *et al.* (2010b)
Use of visulisation in active learning	Section 6.4	Mac Namee *et al.* (2010)
New method to seed active learning	Chapter 7	Hu *et al.* (2010c)
Understanding of the reusability problem	Chapter 8	
Best classifier in active learning	Chapter 8	

Chapter 2

Supervised Learning and Text Classification

Machine learning is inherently a multidisciplinary field which draws on concepts and results from many fields, including probability and statistics, artificial intelligence, philosophy, psychology, information theory, cognitive science and other fields. This chapter introduces notations and provides a brief introduction to the formalisms most pertinent to the work presented in this thesis. More specifically, Sections 2.1 sketches basic principles and concepts of supervised learning with a focus on classification learning. Section 2.2 describes text classification. Section 2.3 provides a brief overview of common approaches to text classification. Finally, methods to evaluate the performance of the text classification systems are presented.

2.1 Basic Concepts

In this section, a formal definition of the elements involved in supervised machine learning is given. We have revised some of the definitions and adapted them to the purpose of this thesis, with the aim of clarifying the classification task and setting a base terminology for further discussion. Therefore, the notation and definitions exposed in this section are the result of some consolidation of the most used formalisms we can find in supervised learning literature.

The goal of supervised learning is to find a function $g : X \rightarrow Y$ which maps an example $x \in X$ to its output value $y \in Y$, as shown in Definition 1.

Definition 1. ***(Example, Output)*** *An example $x \in X$ represent an input object in the data. X is the set of all possible examples in the input space where $X = \{x_1, \ldots, x_i, \ldots, x_N\}$. The output $y \in Y$ represents an output value of an output space and Y is the set of all possible values.* ♣

Two types of learning problems are often defined depending on the output values Y: *regression learning* where $Y = \mathbb{R}$, *classification learning* where $Y = C$ (C is a set of classes and $C = \{c_1, \ldots, c_j, \ldots, c_M\}$). *Classification* is the process to output a value that matches an example to a class. The focus of this thesis is on classification learning. The output values $y \in Y$ in classification learning are called *classes* or *labels*. When applied to text classification tasks, an example is a text document, such as a recipe document. The classes could be the types of recipes, for example $Y = \{starter, maincourse, dessert\}$.

The function g is an important factor in a machine learning system. To design a machine learning approach for a specific problem, it is a key step to select an appropriate function (Mitchell, 1997). The function is called a *model* or a *classifier* in classification learning as defined in Definition 2. A binary classifier assigns a positive ($+1$) or negative class (-1) to each example; a multi-class classifier assigns a class from a set of classes C ($|C| > 2$) to an example; and a multi-label classifier assigns a subset of the set of classes to an example, that is, it assigns more than just one class to an example.

Definition 2. ***(Classifier)*** *A classifier is a function $\Psi : X \to C$ that maps examples to assignment classes. For example, for the binary classifier $C = \{-1, +1\}$, so $\Psi : X \to \{-1, +1\}$.* ♣

Usually, an example is characterised by a vector of *features* which is a vector of real values with a dimension for every feature in the feature space: $\vec{x} \in \mathbb{R}^k$ and $\vec{x} = (f_1(x), \ldots, f_k(x))$. A feature is any item that can be considered a characteristic of an example by a classifier. It is important to identify which items will be features for the example. More details about feature representations of text documents can be found in Section 2.2.1.

2.2 Text Classification

Text classification (a.k.a. *text categorization* – TC) (Yang, 1999) is the task of assigning predefined categories to texts based on the contents of the documents. Text classification has an important role to play in the field of natural language processing or other text-based knowledge applications, especially with the recent explosion of readily available text data – such as electronic news articles, digital libraries, and blogs. One example is spam filtering (Drucker *et al.*, 1999) which attempts to sift through a user's incoming emails and identify those that are unsolicited, unwanted or inappropriate – those that are considered 'spam' to the user. Another example is sentiment analysis (Pang *et al.*, 2002) which aims to determine the overall sentiment (positive or negative) expressed in a document, for example, the classification of online product reviews into those that are positive or negative about a product.

Text classification problems can be solved by applying supervised learning algorithms to train classification models with a collection of previous examples of the problem in question, for which the correct classifications (or labels) are known. These models can then be used to predict the labels of unlabelled documents (Dumais *et al.*, 1998; Joachims, 1999; Nigam *et al.*, 1999; Soucy & Mineau, 2001; Vidhya.K.A & G.Aghila, 2010; Yang & Liu, 1999).

A text classification system may be built up from the following components:

- Text representation: to convert documents into a set of features that can be handled by classification algorithms;
- Feature reduction: to reduce the number of features to make classification algorithms efficient which can be implemented by methods including dimension reduction (Davy & Luz, 2007b) and feature selection (Abe & Kudo, 2006; Wiratunga *et al.*, 2006; Yang & Pedersen, 1997);
- Classifier training: to build up an autonomous classifier by using supervised learning algorithms;
- Prediction: to generate labels or classifications for new documents by using

the trained classifier.

Comparisons among different text classification techniques can be found in Yang & Liu (1999) and Demšar (2006).

2.2.1 Text Representation

In the representing step, documents are often tokenised first and then presented as feature vectors (a.k.a *Vector Space Model* (VSM) or *Bag-Of-Words* (BOW)) (Raghavan & Wong, 1986; Salton *et al.*, 1975; van Rijsbergen, 1979) where a subset of distinct words occurring in the given documents are used as features. Words that are too common in the language are thought to be meaningless terms which are known as *stop words* include determiners, prepositions, auxiliaries, and so on. Stop words are usually removed. Words that occur in very few documents may also be removed. Stemming analysers such as Porter stemming algorithm (Porter, 1980) can be used to remove the commoner morphological and inflexional endings from words.

The vector space model assumes that each word (or *term*) is a dimension in the feature space. The dimension of the term space is the number of terms t in corpus. Each term t_i in a text or document d_j has an associated weight w_{ij}. One document d_j can be represented as: $\vec{d_j} =< w_{1j}, w_{2j}, \ldots, w_{tj} >$. A collection of n documents can be expressed by a Term×Document Matrix (TDM) where w_{ij} is the weight or frequency of term i in document j, while $|T|$ and $|D|$ denotes the number of terms and documents, respectively.

$$\mathbf{TDM} = \begin{bmatrix} & t_1 & t_2 & \dots & t_t \\ d_1 & w_{11} & w_{21} & \dots & w_{t1} \\ d_2 & w_{12} & w_{22} & \dots & w_{t2} \\ : & : & : & & : \\ : & : & : & & : \\ d_n & w_{1n} & w_{2n} & \dots & w_{tn} \end{bmatrix}$$

Different weighting schemes have been used (Salton & Buckley, 1988; Soucy & Mineau, 2005) to calculate the weight w_{ij} including a binary weight schema and more complex term-weighting scheme. With the binary weight schema, if a term t_i occurs in a document d_j then the weight w_{ij} is 1, otherwise $w_{ij} = 0$. *Term Frequency* (TF) gives each feature a weight proportional to the number of times it occurs within a given document. In most situations of term frequency schemes, tf is normalised with some normalisation techniques to prevent a bias towards longer documents. For example:

$$f_{ij} = frequency\ of\ term\ i\ in\ document\ j$$

$$tf_{ij} = f_{ij}/\max_i\{f_{ij}\}$$

Document Frequency (DF) is also employed in order to construct more sophisticated weighting mechanisms for texts. Document frequency of term i (df_i) means the number of documents containing term i. Often *Inverse Document Frequency*

(IDF) is used. The rationale behind IDF is that features that rarely occur over collections of documents are valuable, and that therefore, the IDF of a feature is inversely proportional to the number of documents that it appears in. One example is shown as following:

$$idf_i = \log_2(N/df_i)$$

TF·IDF (Baeza-Yates & Ribeiro-Neto, 1999; Salton & Buckley, 1988) is one of the most common weighting scheme used where the term frequency is weighted by the inverse document frequency, that gives a higher weight to infrequent terms in the dataset:

$$w_{ij} = tf_{ij} \times idf_i = tf_{ij} \times \log_2(N/df_i)$$

2.3 Approaches to Text Classification

Various models have been described in the literature on machine learning, such as those listed below:

- Probabilistic classifiers, such as Naïve Bayes classifier (McCallum & Nigam, 1998a; Vidhya.K.A & G.Aghila, 2010);
- Decision trees (Mitchell, 1997; Quinlan, 1992);
- Margin classifiers, such as Support Vector Machines (Cortes & Vapnik, 1995; Joachims, 1997; Vapnik, 1995);
- k-Nearest Neighbour classifiers (Cover & Hart, 1967; Duda *et al.*, 2000);
- Neural networks (Martín-Valdivia *et al.*, 2003);

- Boosting and ensembles (Schapire & Singer, 2000);
- Other algorithms such as maximum entropy modeling (Greiff & Ponte, 2000) and logistic regression (Gey, 1994).

Among them, three most common approaches used for text classification tasks including the k-Nearest Neighbour classifier, Support Vector Machines and the Naïve Bayes classifiers are described in more detail in the following sections.

2.3.1 k-Nearest Neighbour Classifiers

Nearest Neighbour (NN) algorithms are well-known and intensively used methods. The nearest neighbour classification is straightforward and intuitive: given a set of labelled training examples, a new unlabelled target example is assigned to the same label as its nearest neighbour which is identified using a similarity measure. It is often more accurate to take more than one neighbour into account, leading to the more common nearest neighbour classifier namely *k-Nearest Neighbour* (k-NN) where k nearest neighbours are used to determine the class of the given target example (Cover & Hart, 1967; Duda *et al.*, 2000).

The determination of the similarity between the given target and its neighbours is mostly based on distance measures. Two typical distance functions are the absolute distance (shown in Equation 2.1 where $r = 1$) and the Euclidean distance (shown in Equation 2.1 where $r = 2$).

$$d(A, B) = (\sum_{i=1}^{n} |a_i - b_i|^r)^{1/r} \tag{2.1}$$

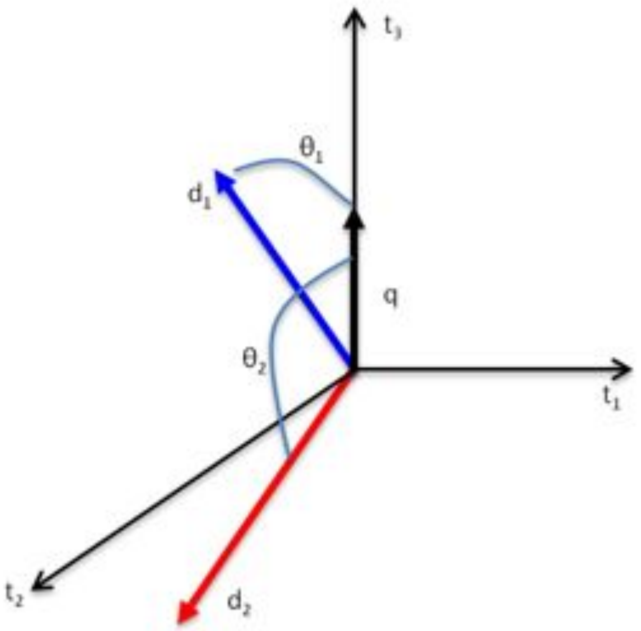


Figure 2.1: Cosine similarity. Adapted from Raymond J. Mooney's talk about "Text Categorization".

Cosine similarity (Baeza-Yates & Ribeiro-Neto, 1999) is commonly used in text classification to measure the similarity of two document vectors based on the cosine of the angle between two vectors as shown in Figure 2.1. The similarity between document d_1 and the target document q is measured by the cosine of the angle θ_1. The same with the angle θ_2 between d_2 and q. The cosine similarity between documents d_j and d_k is defined in Equation 2.2.

$$CosSim(d_j, d_k) = \frac{\vec{d_j} \cdot \vec{d_k}}{|\vec{d_j}| \cdot |\vec{d_k}|} = \frac{\sum_{i=1}^{t}(w_{ij} \cdot w_{ik})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{ik}^2}} \tag{2.2}$$

Once the k-NN classifier determines the k neighbours (training examples) that are closest to the target example, a voting scheme is used to decide the class prediction of the target example. Majority voting is the most commonly used scheme. However in simple majority voting, the k neighbours contribute equally to the decision of the prediction of the given target. This can be problematic. For example, in a sparse space, some of the neighbours of an target example may be far away and have little or no effect on the target example. In order to overcome this limitation, Weighted k-Nearest Neighbour (WKNN) technique (Hechenbichler & Schliep, 2004; Kibler & Aha, 1987), is proposed which is based on the idea that the close neighbours should get a higher weight in the decision than far ones.

An overview of k-NN classifiers can be found in Cunningham & Delany (2007). The main computation in the k-NN algorithm is the sorting of training examples in order to find the k nearest neighbours for the target example. Several algorithms (Arya *et al.*, 1998; Connor & Kumar, 2010; Garcia *et al.*, 2008) have been proposed in order to reduce the computational burden of k-NN classifiers. The advantage of the k-NN algorithm includes its simplicity, flexibility to incorporate different data types and adaptability to irregular feature spaces. Even with its simplicity, the results obtained by applying k-NN algorithm are competitive, as shown by Yang & Liu (1999).

2.3.2 Support Vector Machines

The Support Vector Machine (SVM) (Vapnik, 1995) is probably the most prominent approach to maximum margin classification which is essentially specified by a separating hyperplane in the multi-dimensional input space $\mathbb{R}^k$ given by the feature representation $\vec{x}$ of the example x. The best separating hyperplane is the one that represents the largest separation – called *margin*. The distance from it to the nearest training example on each side is maximised. The hyperplane is defined as

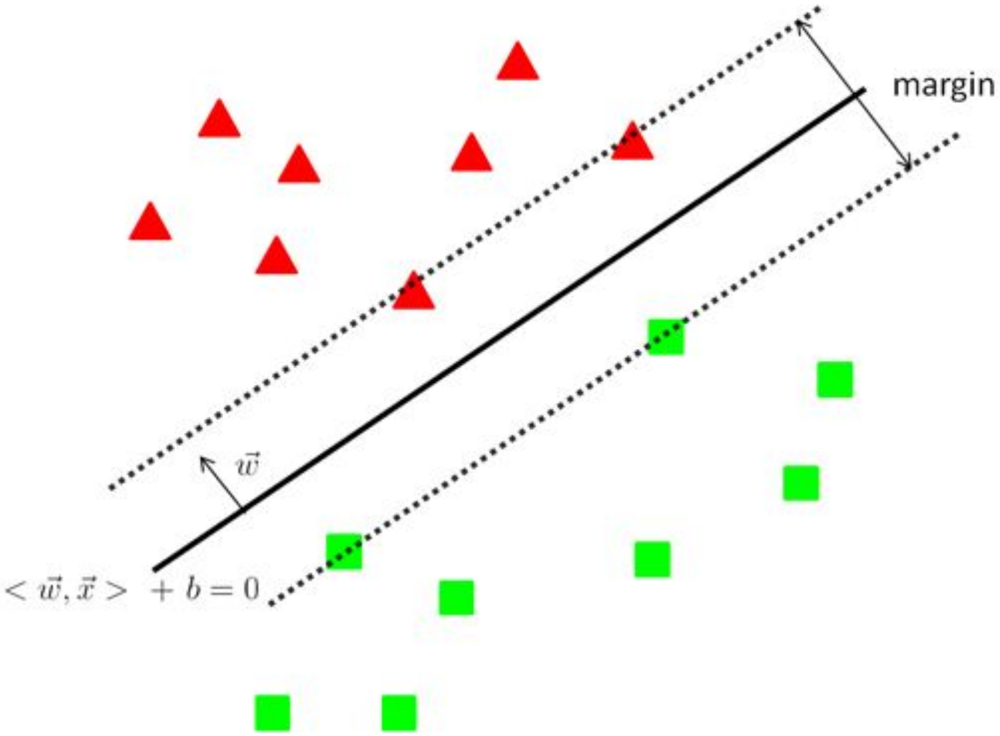


Figure 2.2: SVM for a simple classification scenario: The solid line represents the separating hyperplane. Points on the dashed lines are the support vectors. The figure is adapted from (Tomanek, 2010).

in Equation 2.3 where $\vec{w}$ is a weight vector in $\mathbb{R}^k$ defining the orientation of the hyperplane and b is an offset ($b \in \mathbb{R}$).

$$< \vec{w}, \vec{x} > \; + b = 0. \tag{2.3}$$

The basic idea behind SVM is to find those examples (*support vectors*) that delimit the widest frontier between positive and negative examples in the feature space as shown in Figure 2.2 (see Burges (1998) for a detailed introduction to SVM). Support vectors are examples which are closest to the hyperplane. The width of the classification border is known as the *hyperplane margin*. The SVM classifier is given by Equation 2.4. The examples for which $< \vec{w}, \vec{x} > \; + b > 0$ are classified as positive examples, or negative otherwise.

$$f_{\vec{w},b}(x) = sgn(< \vec{w}, \vec{x} > \; + b) \tag{2.4}$$

SVM classifiers can deal with non-linearly separable data by mapping the feature representation $\vec{x}$ of example x with a non-linear mapping function $\Phi : \mathbb{R}^k \rightarrow \mathcal{H}$ into a higher-dimensional feature space $\mathcal{H}$ where the separability between examples may be easier. This is done using a *kernel function* $K(\vec{x}, \vec{y}) =< \phi(\vec{x}), \phi(\vec{y}) >$ leading to a reformulation of the SVM classifier into Equation 2.5 where s is the number of support vectors which have non-zero α_i values. Well-known kernel functions include the linear kernel, the polynomial kernel, Radial Basis Functions (RBF) , and sigmoid kernels.

$$\hat{\Phi}(x) = \sum_{i=1}^{s} \alpha_i y_i K(x, w_i) + b. \tag{2.5}$$

While SVM classifies are formulated for binary classification problems, there are ensemble approaches to multi-class classification with SVMs, such as the *one-against-rest* (also namely *one-against-all*) (Bottou *et al.*, 1994), *one-against-one* and *Error Correcting Output Codes* (ECOC) (Morelos-Zaragoza, 2006) . A unified approach of reducing multi-class to binary for margin classifiers has been proposed in Allwein *et al.* (2000). A comparison of methods for multi-class SVM classifiers can be found in Hsu & Lin (2002).

Support Vector Machines have been applied successfully in many text classification tasks since most text classification problems are linearly separable and SVMs are robust in high dimensional space and robust with sparse data (Joachims, 1997).

2.3.3 Naïve Bayes Classifiers

The Naïve Bayes classifier is a popular supervised learning technique for text classification and has been found to perform surprisingly well despite its simplicity (Domingos & Pazzani, 1997; Friedman *et al.*, 1997; Langley *et al.*, 1992; Lewis, 1998; Lewis & Ringuette, 1994; McCallum & Nigam, 1998a). For a good survey of using Naïve Bayes classifiers in text classification see Vidhya.K.A & G.Aghila (2010).

The underlying theorem for Naïve Bayes classifiers is Bayes' Law as in Equation 2.6 which is based on the assumption that all features are conditional independent. A probabilistic model that embodies the assumption is posited and training examples are used to estimate the parameters of the proposed model. The classification on a new examples is performed by selecting the class that is most likely to have generated the example.

$$P(y|x) = \frac{P(y)P(x|y)}{P(x)} \tag{2.6}$$

In a dataset, every document has the same probability, so $P(x)$ is a constant which can be eliminated from Equation 2.6. Based on the *Naïve Bayes assumption* that all features are conditional independent, the *Naïve Bayes* (NB) model is formulated as in Equation 2.7. This is used in text classification to determine the probability that a document x is of class y just by looking at the frequencies of words in the document.

$$P(y|\vec{x}) \propto P(y, \vec{x}) = P(y) \prod_{j=1}^{k} P(x_j|y). \tag{2.7}$$

The Naïve Bayes classifier exists in different versions, depending on how examples are represented. One version is called *multi-variate Bernoulli* model (Koller & Sahami, 1997; Larkey & Croft, 1996) in which an example is represented as a binary vector of term occurrences. Another is the multinomial model in which an example is represented as a vector of term counts (Mitchell, 1997). Descriptions of the differences of these two models can be found in McCallum & Nigam (1998a), Schneider (2004), and Schneider (2003). Several work shows that the multinomial model usually performs better than the multi-variate Bernoulli model (Eyheramendy *et al.*, 2003; McCallum & Nigam, 1998a).

There was work that focused on the analysis of the limitations of the Naïve Bayes models (McCallum & Nigam, 1998a; Rennie *et al.*, 2003). Airoldi *et al.* (2004) proposed an extension to the widely-used multinomial model for texts which allows better modelling of frequent words.

2.4 Performance Measures

This section describes measures and methods to evaluate the performance of classifiers. Lewis (1991) supplied a review on how evaluation is carried out in text classification systems. The evaluation of a text classification system is usually based on test examples that have been already labelled by human experts (Schütze *et al.*,

Table 2.1: Contingency table for binary classification scenario.

		true class		
		p	***n***	**total**
predicted class	***p’***	TP	FP	P’
	n’	FN	TN	N’
	total	P	N	

2006).

Table 2.1 shows a *contingency table* (or *confusion matrix*) illustrating the outcomes for a binary classification problem where one class is known as *positive* and the other as *negative*. If the outcome from a prediction is positive and the actual label is also positive, then it is called a *true positive* (TP); however if the actual label is negative then it is said to be a *false positive* (FP). Conversely, a *true negative* (TN) has occurred when negative examples are correctly labelled as negative, and *false negative* (FN) is when positive examples are incorrectly labelled as negative.

Given the numbers from the contingency table, several performance measures can be calculated, such as *accuracy* (Equation 2.8), *error rate* (Equation 2.9), *true negative rate* (TNRate, or *specificity*, Equation 2.10) and *false positive rate* (FPRate, Equation 2.11). The true negative rate measures the proportion of negative examples that are correctly labelled. The false positive rate measures the proportion of negative examples that are misclassified as positive. Besides these measures, *precision* and *recall* from information retrieval are adopted and commonly used for evaluating a text classification system. Precision is the ratio of correct labelling of positive examples (or target examples) divided by the total number of the predictions that are positive, shown in Equation 2.12. Recall is defined to be the ratio of correct labelling positive examples (or target examples) divided by the total number of examples whose actual labels are positive in the dataset, shown in Equation 2.13. The *true positive rate* (TPRate, or *sensitivity*) which measures the proportion of positive examples that are correctly labelled is calculated in the same way as recall as in Equation 2.13.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.8}$$

$$Err = 1 - ACC \tag{2.9}$$

$$TNRate = Specificity = \frac{TN}{N} = \frac{TN}{FP + TN} \tag{2.10}$$

$$FPRate = 1 - Specificity = \frac{FP}{N} = \frac{FP}{FP + TN} \tag{2.11}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.12}$$

$$Recall = TPRate = Sensitivity = \frac{TP}{TP + FN} \tag{2.13}$$

In a non-binary classification problem, micro-averaging and macro-averaging (Lewis, 1992) are introduced to calculate precision and recall. Micro-averaged values are calculated by constructing a global contingency table, and then calculating precision (Equation 2.14) and recall (Equation 2.15) using these sums. In contrast, macro-averaged scores are calculated by first calculating precision and recall for each class and then taking the average of these, as shown in Equation 2.16 and 2.17

respectively.

$$P_{micro} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FP_i)} \tag{2.14}$$

$$R_{micro} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FN_i)} \tag{2.15}$$

$$P_{macro} = \frac{1}{|C|}\sum_{i=1}^{|C|} \frac{TP_i}{TP_i + FP_i} \tag{2.16}$$

$$R_{macro} = \frac{1}{|C|}\sum_{i=1}^{|C|} \frac{TP_i}{TP_i + FN_i} \tag{2.17}$$

It is usually beneficial to have a single measure assessing the system performance. One of the most used measures for the overall performance is F1 measure which is defined as the harmonic mean of precision and recall, shown in Equation 2.18. Another overall measurement, *Precision-Recall BreakEven Point* (PRBEP) is defined as the precision and recall value at which the two are equal.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \tag{2.18}$$

Provost *et al.* (1998) suggested that using measures such as accuracy can be misleading and proposed to use Receiver Operator Characteristic (ROC) curves to present results for binary classification problems. A receiver operating characteristic curve is a plot of the true positive rate on the y-axis against the false positive rate on the x-axis which shows how the number of correctly classified positive examples varies with the number of incorrectly classified negative examples. In order to compare ROC curves of different classifiers, one often calculates the *Area Under the receiver operating characteristic Curve* (AUC) (Hanley & McNeil, 1982) .

When comparing multiple classifiers, usually multiple runs of the experiments are launched with a different partition of the training set and the test set to reduce the possible bias introduced by the splitting of training set and test set. *n-fold cross-validation* strategy (Chang *et al.*, 1992; Devijver & Kittler, 1982; Kohavi, 1995; Manning & Schutze, 1999) is used most frequently which randomly divides the data collection into n sets of equal size. At each turn, one set is used for testing and the rest for training the system. The final result is an averaged measurement of the experiment over every partition made.

Statistical tests are used to decide whether the performance difference between multiple classification algorithms are significant or not. For example, Yang & Liu (1999) compared five text classification methods by using a set of designed significance tests. Dietterich (1998) gave a review of five approximate statistical tests for determining whether one classification algorithm is better than another one on a particular classification problem and suggested that cross validation t-test is very powerful. Demšar (2006) recommended a set of simple non-parametric tests for statistical comparisons of classifiers with results that are not normally distributed: the Wilcoxon signed-rank test (Wilcoxon, 1945) for comparison of two classifiers and the Friedman test (Friedman, 1940) for comparison of more classifiers over multiple datasets.

2.5 Conclusions

In this chapter, we have outlined basic concepts and approaches to classification problems. Classification is one core area of supervised learning research. The goal of classification learning is to put examples into classes based on a given collection of labelled examples – a training set. There has been significant research into text classification where examples to be classified are presented in textual forms. Typical text classification applications include spam filtering, topic classification and sentiment classification. There are a number of approaches to deal with classification problems. Among them, k-Nearest Neighbour classifiers, support vector machines and Naïve Bayes classifiers are very popular. Measures including accuracy, precision, recall, and AUC are usually used to measure the performance of text classification systems.

In order to build a text classification system, a labelled training dataset would be required. It is easy to get the articles themselves but hard to label them because of the huge amount of documents involved and the time requirement. Usually human experts need time to read those articles and label them. On the other hand, not only reading but also judgement about the articles are needed to label them. This would make the labelling process time consuming and expensive. Active learning is a machine learning technique which can be used to build labelled datasets with minimum human labelling effort. Our objective is to apply active learning to the problem of generating labelled textual datasets. The next chapter gives a literature review on active learning with emphasis on the selection strategies design.

Chapter 3

Active Learning

As discussed in Chapter 2, when using supervised learning methods to solve learning problems, a set of examples labelled with their actual classes is required to train a model. The success or failure of such learning systems is largely determined by the datasets used to train them. Without a good dataset it is difficult to build a quality classifier. Unfortunately, manually generating collections of labelled examples is typically time-consuming and expensive since it usually involves expensive experts such as doctors or engineers. This can be a real barrier to the creation of classification systems, as often the time or money is not available to build a labelled training set. Fortunately, this is not an insurmountable problem. Active learning is a machine learning technique that can be used to select only the most informative examples for labelling and help to reduce the labelling effort.

3.1 Basic Concepts and Problem Settings in Active Learning

As was mentioned previously in Section 2.1, in traditional supervised learning, the learner is given some training examples and builds a classification model. The training examples are usually selected by random sampling. This way is referred to as *passive learning* (Tong, 2001). However, in many text classification tasks, labelled training documents are often expensive to obtain, while unlabelled documents are readily available in large quantities. For example in document relevancy filtering (Xu *et al.*, 2007) although there are a large number of potential unlabelled documents available, labelled documents are expensive to obtain because human evaluation is required in order to ascertain relevance.

Active Learning (AL) (see Settles, 2009, for a review) is an iterative 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). Active learning first garnered serious research attention in the 1980s (Angluin, 1988) and since has remained a vibrant research area. Active learning is widely used in situations where there are vast amounts of unlabelled data available (e.g. astrophysical data (Schneider, 2009), image retrieval (Dagli *et al.*, 2005; Gosselin & Cord, 2005; Hoi *et al.*, 2009b; Tong & Chang, 2001), natural language processing (Tomanek & Hahn, 2009; Zhu & Hovy, 2007) and text classification (McCallum & Nigam, 1998b; Novak *et al.*, 2006)) 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)). Several recent applications of active learning are summarised in Table 3.1.

Table 3.1: Some successful applications of active learning.

Domain	**Description**
bioinformatics	drug discovery (Warmuth *et al.*, 2003), medical image classification (Hoi *et al.*, 2006b), cancer classification (Liu, 2004), mining of cell assay images (Cebron & Berthold, 2005), plankton recognition (Luo *et al.*, 2005), medical subject headings assignment (Sohn *et al.*, 2008), time-series gene expression analysis (Singh *et al.*, 2005)
natural language processing	named entity recognition (Laws & Schütze, 2008; Shen *et al.*, 2004), word sense disambiguation (Imamura *et al.*, 2009; Zhu & Hovy, 2007), parse selection (Baldridge & Osborne, 2003), spoken language understanding (Tur *et al.*, 2005), frame assignment (Ghayoomi, 2010)
detection	outlier detection (Abe *et al.*, 2006), network intrusion detection (Li & Guo, 2007), malicious code detection (Moskovitch *et al.*, 2007), rare category detection (He & Carbonell, 2007), mine detection (Juszczak & Duin, 2003)
robotics	robot manipulation (Morales *et al.*, 2004)
astrophysics	anomaly and rare-category detection (Pelleg & Moore, 2004)
texts	classification (McCallum & Nigam, 1998b; Nigam & McCallum, 1998), texture segmentation (Juszczak & Duin, 2003), spam filtering (Sculley, 2007; Segal *et al.*, 2006), labelling textured data (Turtinen & Pietikäinen, 2005)

information extraction	extracting attribute-value pairs from product descriptions (Probst & Ghani, 2007); extracting noun phrases that belong to a predefined set of semantic classes (Jones *et al.*, 2003)
information retrieval	relevance feedback (Dagli *et al.*, 2006; Xu *et al.*, 2007), music retrieval (Mandel *et al.*, 2006), active information retrieval framework (Loureiro & Siegelmann, 2005)
software engineering	automatic classification of software behavior (Bowring *et al.*, 2004)
speech recognition	select a good training-data subset for transcription (Lin & Bilmes, 2009; Yu *et al.*, 2009), classify voicemail messages (Kapoor *et al.*, 2007b)
recommender systems	minimize the number of requests for user evaluations (Teixeira *et al.*, 2002)
computer vision	image classification (Li *et al.*, 2004b), video annotation (QI *et al.*, 2006; Yan *et al.*, 2003), visual object detection (Abramson & Freund, 2005), image segmentation (Ma *et al.*, 2006), human motion capture (Cooper *et al.*, 2007), object category recognition (Kapoor *et al.*, 2007a), object detection (Nguyen *et al.*, 2009), image retrieval (Cord *et al.*, 2007; Dagli *et al.*, 2005), optical character recognition (Monteleoni & Kaariainen, 2007)

There are three main forms that have been considered in the active learning literature: (i) active learning with membership queries, (ii) stream based active learning, and (iii) pool-based active learning.

In *active learning with membership queries* (Angluin, 1988), on each trial the learner constructs an example in the input space and requests its label from the oracle. This approach was originally used in concept induction (Angluin, 1988) and involved learning to identify an unknown concept drawn from a finite hypothesis space using queries to gather information about the unknown concept. It has been used to predict the target position of a robot hand given a set of joint angles (Cohn, 1996) and to classify handwritten characters (Baum & Lang, 1992). The main advantage of active learning with membership queries is that it can work in situations when unlabelled data is unavailable. However, Baum & Lang found that examples generated by the algorithm are not meaningful and recognizable by the human experts. This approach is almost impossible for text classification because the documents constructed by the algorithm are implausible examples which have no meaningful contents.

In *stream based active learning* (Freund *et al.*, 1997), queries are based on filtering a stream of unlabelled examples rather than on creating artificial examples. The learner is presented with a stream of unlabelled examples and chooses whether or not to query the oracle for the label for each example in the stream. Stream based active learning is predominately used in time series data such as video sensor data (Saunier *et al.*, 2004). The advantages of stream based active learning is that it can deal with complex and noisy data (in the case of temporal evolving data) and it can be used in dynamic and online learning scenarios. Sculley (2007) investigated the use of stream based active learning method for spam filtering where the filter was exposed to a stream of messages. Stream based approaches have the disadvantage of the learner not being able to access all unlabelled examples when trying to select the most informative ones for which labels will be requested.

In text classification, usually a large collection of unlabelled documents are available so the most common form of active learning used is the *pool-based* approach (Lewis & Gale, 1994; Nigam & McCallum, 1998; Tong & Koller, 2001). Pool-based active learning assumes that the learner has access to a large pool of unlabelled examples from the beginning of the process, and this is the scenario considered in this work. Although pool-based active learning is the most common form of active learning, one disadvantage is that pool-based active learning deals with static or non-changing datasets which can cause difficulties in dynamically changing, online environments. We will discuss the active learning process and commonly used pool-based active learning methods in detail in the remainder of this section.

3.1.1 Pool-Based Active Learning Process

A conceptual view of pool-based active learning can be modelled as a quintuple, <$\mathcal{S}$, $\mathcal{O}$, $\mathcal{L}$, $\mathcal{U}$, $\mathcal{SC}$> (Baram *et al.*, 2004). A small set of training examples, $\mathcal{L}$, that are labelled by an oracle, $\mathcal{O}$, is used to initialise a selection strategy, $\mathcal{S}$. The selection strategy 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 and then presenting the top most informative unlabelled examples to the oracle $\mathcal{O}$ for labelling. A *batch size* b determines the number of examples to be selected in each active learning iteration. It has been suggested that a smaller batch size leads to a sharper increase in performance (Schohn & Cohn, 2000), however, batch

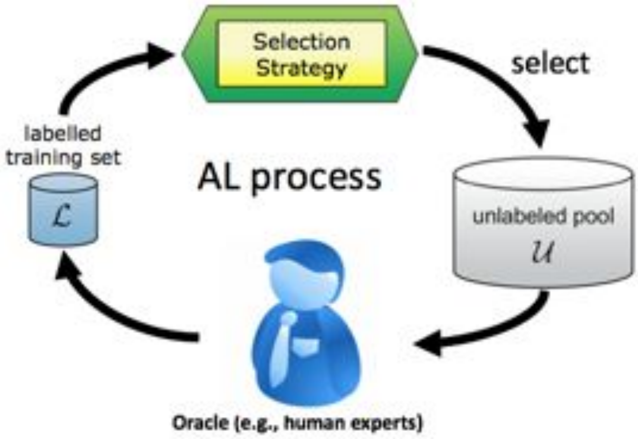


Figure 3.1: Pool-based active learning framework.

mode active learning using larger values of b is more efficient (Hoi *et al.*, 2006a). The labelled examples are then 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 criteria ($\mathcal{SC}$) is reached – for example the oracle exceeds a label budget, or labelling further examples is not deemed sufficiently informative.

A framework of a generic pool-based active learning system is shown in Figure 3.1 (see Algorithm 1 for the pseudo-code). The resulting manually labelled set can be used to train a classifier or infer the labels for the remaining of the unlabelled examples. Typically, the manually labelled set is used to build a classifier.

There are three significant issues of concern in active learning. First, a technique is required to choose a small initial training set to seed the active learning process. Second, a selection strategy is required to select the examples that will be labelled throughout the active learning process. These should be the examples for which labels will prove most informative as the training process progresses. Third, criteria

```
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
while $\mathcal{SC}$ 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.

must be established to determine when the active learning process should stop. These three are discussed below with a focus on the selection strategy since it is at the heart of the active learning process.

3.2 Initial Training Set Construction

Pool-based active learning process begins with a small set of initially labelled examples, $\mathcal{L}$. A technique is required to choose this set to seed the active learning process. In most of the active learning applications, this is done randomly (Lee *et al.*, 2009; Novak *et al.*, 2006; Schohn & Cohn, 2000; Thompson *et al.*, 1999). However, the problem of randomly picked initial training sets has been addressed in Zhu *et al.* (2008b). It was argued that in real-world applications, the randomly sampled initial training set does not have the same prior data distribution as that of the whole dataset because of the small size of initial training set. A randomly selected initial training set works well only when the chances are good that it has the same prior data distribution as the whole dataset.

To populate the initial training set in a more targeted way, clustering techniques can be used. According to Nguyen & Smeulders (2004) the most representative examples in a collection are likely to be those at the centres of clusters and these should be used as initial training examples to seed the active learning process. In the active learning literature there are some examples that take this approach, typically using k-Means (Kang *et al.*, 2004; Zhu *et al.*, 2008b), k-Medoids (Nguyen & Smeulders, 2004), or spectral clustering (Dasgupta & Ng, 2009). In a variation on the k-Means approach for initial training set selection, artificial examples built from the virtual centroids, named *model examples*, are also added to the initial training set (Kang *et al.*, 2004). This approach is named *KMeans+ME* and leads to an initial training set twice the size of that created when using just k-Means. Kang *et al.* suggested that with a well selected initial training set, the active learner can reach high performance faster with fewer queries. The same conclusion was supported by the experiments of QI *et al.* (2006) where an initial training set for active learning video annotation was constructed using time-constraint clustering.

3.3 Selection Strategies

Active learning selection strategies can be categorised into three approaches: *exploitation based selection strategies*, *exploration based selection strategies*, and strategies that use a combination of both exploitation and exploration. The remainder of this section will discusses methods of each of these types in detail.

3.3.1 Exploitation Based Selection Strategies

Exploitation based selection strategies build a classifier using those examples labelled by the oracle so far in the active learning process and base the selection of examples for labelling on the output generated by this classifier when it is used to classify all of those examples remaining in the unlabelled pool (Cebron & Berthold, 2008).

Uncertainty sampling (Lewis & Gale, 1994) is the most widely used exploitation based selection strategy in text classification (Lewis & Gale, 1994; Raghavan *et al.*, 2006; Segal *et al.*, 2006; Tong & Koller, 2001). The advantages of the uncertainty sampling approach include its simplicity and fast execution speed.

Typically in uncertainty sampling a ranking classifier (e.g. k-Nearest Neighbour, Naïve Bayes or Support Vector Machines) is trained using the examples labelled by the oracle so far and this classifier is then used to classify the remaining unlabelled examples. Using the output of the ranking classifier as a measure of *uncertainty*, those examples for which classifications are least certain are selected for labelling by the oracle. More details of classifiers used in active learning can be found in Section 3.3.1.1.

Uncertainty sampling focuses on labelling examples near the current classification boundary so as to fine-tune this boundary. Lewis & Gale (1994) utilised a probabilistic classifier to produce a confidence score $P(C|x)$ for an example x when it is predicted as class C and queried the oracle for labels for examples whose confidence score is closest to 0.5. Uncertainty can also be measured using entropy when probabilistic classifiers are employed (see Hwa, 2004, for an example). Uncertainty sampling can work with non-probabilistic classifiers by modifying the output of the classifier to have a probabilistic output (Fujii *et al.*, 1998; Lindenbaum *et al.*, 2004). Another approach measures uncertainty by how far the examples are away from the classification boundary (the SVM unit vector obtained from the training data), such as the method proposed in Tong & Koller (2001).

Instead of using the direct classifier output as the measurement of uncertainty, classifier confidence measurements have been used in *confidence based active learning* (Li & Sethi, 2006a). Detailed discussion on confidence based active learning will be in Section 3.3.1.2.

Interesting extensions to typical uncertainty sampling include fast uncertainty sampling and historical uncertainty sampling. Segal *et al.* (2006) proposed fast uncertainty sampling to improve the efficiency for labelling large email datasets. In most of the uncertainty sampling algorithms, just the current prediction is used for the computation of uncertainty. Davy & Luz (2007a) extended uncertainty sampling by using history information and proposed two methods that incorporated historical predictions into uncertainty sampling. Historical predictions for a particular unlabelled example were stored for all iterations of active learning to date. In the first method namely *History Uncertainty Sampling* (HUS), the uncertainty of an example was defined as the sum of the uncertainty of the last k predictions. In the second method namely *History KLD*, the past k predictions were thought of as the output of a committee of size k and the uncertainty was measured as the disagreement among committee members using Kullback-Leibler divergence to the mean (Pereira, 1993).

Query-By-Committee (QBC) (H.S.Seung *et al.*, 1992) is another popular exploitation based method which creates a "committee" of classifier variants to classify and select unlabelled examples. Those examples with the biggest classification disagreement among the committee (used as the measure of uncertainty) are then selected for labelling. Common ways of estimating the disagreement are *vote-entropy* (Dagan & Engelson, 1995) and *Kullback-Leibler divergence* (Becker & Osborne, 2005). QBC has been applied in text classification, such as in the work by Liere & Tadepalli (1997).

Active learning with expected error reduction (Roy & McCallum, 2001) can be also viewed as an exploitation based method. The objective of active learning with expected error reduction is to directly minimise the generalization error of the classifier on future test examples. As a result, it does not pick examples close to the classification boundary (as described by Krishnakumar, 2007) which is different from uncertainty sampling. The resulting active learning schemes have the advantage of avoiding the selection of outliers and have been shown to outperform uncertainty based approaches (Roy & McCallum, 2001). However, this kind of approach has very high computational cost and requires a number of optimizations and approximations to be efficient and tractable in practice.

The advantage of exploitation based strategies is that exploiting the current model helps to find an optimal model efficiently. However, a limitation of such strategies is that they rely on the relative correctness or confidence of the current model, which can be a difficulty, especially in the early stages. Exploitation approaches to selection can also suffer from a lack of exploration of the feature space and may not work well in some scenarios – for example in the exclusive OR problems (Osugi *et al.*, 2005).

3.3.1.1 Classifiers in Active Learning

In many selection strategies, especially in exploitation based selection strategies, a classifier is built from a labelled set and then it is used to predict the labels of unlabelled examples in the pool and assign an informativeness measure with each member in the pool. Many selection strategies algorithms have been proposed using support vector machines (Raghavan *et al.*, 2006), logistic regression (Hoi *et al.*, 2006a), Naïve Bayes (Segal *et al.*, 2006), maximum entropy (Zhu *et al.*, 2008a), etc. Among them, we will focus on Naïve Bayes, SVM and k-NN based active learners as they are suitable for large-scale text systems.

Naïve Bayes Based Active Learners As discussed previously in Section 2.3.3, Naïve Bayes classifiers are well-known probabilistic classifiers and became popular because of their simplicity, efficiency and accuracy. Using a Naïve Bayes classifier in active learning is particularly efficient since when training a Naïve Bayes classifier only a single pass over the labelled set is needed to gather term frequencies information and no optimization is needed (Rennie & Rifkin, 2001). Segal *et al.* (2006) used boosted Naïve Bayes classifiers in their work of efficient uncertainty sampling for labelling large email corpora. Roy & McCallum (2001) tried to optimise the true generalization error rate with a Naïve Bayes classifier in active learning using an expected error reduction method. McCallum & Nigam (1998b) used a Naïve Bayes classifier in their QBC selection strategy where they modified the QBC method and applied EM to improve parameter estimates of the Naïve Bayes classifier. After that, Naïve Bayes classifiers became widely used with EM (see Ghani, 2001; Nigam & Ghani, 2000; Nigam *et al.*, 1998, 2000; Probst & Ghani, 2007).

SVM Based Active Learners SVMs have been very successful and are very widely used in active learning. One of the most popular approaches to support vector machine based active learning is proposed by Tong & Koller (2001). They developed their selection strategy based on the analysis of version space. The idea is that the most informative examples are those which can halve whole portions of the version space. Three methods are proposed which are approximations to the querying component that always halve the version space. The first method named 'SIMPLE MARGIN' selects the example closest to the decision hyperplane in the kernel space; namely, the point with the smallest margin. The second method named 'MAXMIN MARGIN' which computes two margins m^+ and m^- for each unlabelled example when it is labelled as positive class and negative class respectively and then chooses to query the examples for which the quantity $\min(m^+, m^-)$ is greatest. The third method named 'RATIO MARGIN' which also computes two margins m^+ and m^- as in MAXIMIN MARGIN but chooses the example to query with the largest value of $\min(\frac{m^+}{m^-}, \frac{m^-}{m^+})$ instead. MAXMIN MARGIN and RATIO MARGIN achieve better performance than SIMPLE MARGIN but the main drawback is their high computation because for each query, two SVMs for each unlabelled example in the pool need to be built.

Since SIMPLE MARGIN is more efficient, it is widely used in SVM based active learning (Moskovitch *et al.*, 2009; Novak *et al.*, 2006), in particular, as it performs quite well on text classification problems (see Godbole *et al.*, 2004). Schohn & Cohn (2000) described an application of using the SIMPLE MARGIN strategy in text classification and found that the performance of the SVM trained with the small set of actively selected documents is better than the SVM trained with the whole dataset. Ertekin (2005) also found that by using active learning the need for training examples for SVM can be significantly reduced, and the learner's classification performance is preserved, even increased in some cases.

Many extensions or variants of basic SVM-based active learners using SIMPLE MARGIN have been proposed. Ertekin (2005) proposed an extension of SIMPLE MARGIN called '*Simple Random Active Learning*'. In Simple Random Active Learning, firstly a small constant number of unlabelled examples are randomly selected then the example closest to the margin among this small set is chosen for querying labels. Raghavan *et al.* (2006) used a method similar to Simple Random Active Learning but focused on extending the traditional active learning framework to include feedback on features in addition to labelling examples. Xu *et al.* (2003) proposed representative sampling as an extension to the SIMPLE MARGIN method. The representative sampling method explores the clustering structure of uncertain documents (documents in the classification margin) and selects clustering centers for labelling. Dasgupta & Ng (2009) used a similar SVM based selection strategy as SIMPLE MARGIN and combined active learning, transductive learning and ensemble learning for sentiment classification.

***k*-NN Based Active Learners** The re-building, re-classifying and re-ranking of the pool can make uncertainty-based active learning very computationally expensive.

As a lazy learner, the k Nearest Neighbour classifier is attractive to active learning as the introduction of new examples to the classifier simply involves adding them to the labelled set, and that so much computation required for classification (e.g. the similarities between all examples) can be pre-computed.

In k-NN based uncertainty sampling, the output of the k-NN classifier can be transformed into a class membership probability estimate where the distribution is based on the distance of the query example to its k nearest neighbours and then the estimate can be used as a measure of uncertainty. Examples of using k-NN classifiers in the active learning process were proposed initially by Hasenjager & Ritter (1998) and Lindenbaum *et al.* (2004). More recent examples include developing recommender systems to minimise the number of requests for user evaluations (Teixeira *et al.*, 2002); investigating dimensionality reduction for active learning with nearest neighbour classifier in text classification (Davy & Luz, 2007b); supervised network intrusion detection method based on Transductive Confidence Machines (TCM-KNN) (Li & Guo, 2007); and building classification systems with a weighted k-nearest neighbour classifier (Cebron & Berthold, 2008).

It is interesting to note that except for a small number of examples as mentioned, the k-NN classifier has not been used popularly in active learning research. This is something that we intend to pursue in this work.

3.3.1.2 Confidence-based Active Learning

As discussed in Section 3.3.1, uncertainty sampling selects the unlabelled example with the maximum uncertainty which implies that the current classifier has the least confidence on its classification of this unlabelled example.

Confidence measures give users more insight into the predictions that a classifier makes. In the case of probabilistic models, the confidence of the classifier is commonly estimated using the posterior probability of its output (Imamura *et al.*, 2009; Li, 2005). Bayesian algorithms usually output useful posterior probabilities as confidence values but when the underlying distribution is not known these values are not meaningful (Melluish *et al.*, 2001). For classifiers such as decision trees, a method for assigning confidences to the predictions of decision trees was described in Schapire & Singer (1999) .

For active learning with nonprobabilistic classifiers such as support vector machines, a popular heuristic for establishing the confidence of estimates and identifying points for active learning is to simply use the distance from the classification boundary (Tong & Koller, 2001). However, Dredze & Crammer (2008) argued that confidence and margin are two distinct properties. Dredze & Crammer presented a method for incorporating confidence into the margin by using an online learning algorithm. Platt (1999) suggested to transform SVM outputs to posterior probabilities by using a sigmoidal function. Li & Sethi (2004, 2006b) suggested using dynamic bin width allocation method to transform the output scores from the SVM to posterior probabilities and then a classification confidence can be assigned to the class of each example using these posterior probabilities. Such a confidence based SVM classifier is applied by Ma *et al.* (2006) to selective object segmentation using active learning. A good discussion about confidence-based active learning methods with SVMs can be found in Li & Sethi (2006a).

For k-NN classifiers, the Transductive Confidence Machine for Nearest Neighbours (TCM-KNN) (Proedrou *et al.*, 2002) follows the transductive approach, where for the classification of every new example it uses the whole training set to infer a rule for that particular example only. In TCM-KNN, confidence is calculated based on a measure named the *strangeness measure* using algorithmic randomness theory (Vovk *et al.*, 1999). The strangeness measure is defined as in Equation 3.1 which is the ratio of the sum of the k nearest distances from the same class (D_{ij}^{y}) to the sum of the k nearest distances from all other classes (D_{ij}^{-y}). TCM-KNN algorithm has been used in active learning for supervised network intrusion detection (Li & Guo, 2007). Another example of using TCM-KNN in active learning can be found in Ho & Wechsler (2003). While Ho & Wechsler focused on the design of the selection strategy and an early stopping criteria, Li & Guo were interested in using the TCM-KNN algorithm for the task of intrusion detection.

$$\alpha_i = \frac{\sum_{j=1}^{k} D_{ij}^{y}}{\sum_{j=1}^{k} D_{ij}^{-y}} \tag{3.1}$$

There has been a lot of work on deriving confidence scores from the k-Nearest Neighbour algorithm. Dasarathy (1995) exploited the concept of *Nearest Unlike Neighbor* (NUN) and developed a measure of confidence based on it. Cheetham (2000) implemented and experimented with multiple methods for determining confidence in a case-based reasoning system. Yao & Ruzzo (2006) extended the k-Nearest Neighbour algorithm and suggested a voting scheme to generate confidence scores that estimate the accuracy of predictions. Roy & Madhvanath (2008) explored the adaptive k-NN classification strategy (Li *et al.*, 2004a) and confidence measure in the context of skewed distributions of training examples. To attach confidence to classification scores, Delany *et al.* (2005a) proposed five basic confidence measures that can be used with k-NN classifiers and showed that an aggregate of these is particularly effective. The use of aggregate measures is also supported by the work of Cheetham & Price (2004) who presented a similar result, using different measures.

3.3.2 Exploration Based Selection Strategies

Exploration based selection strategies pick representative examples from dense regions of the example space instead of focusing on examples closest to the classification boundary. Exploration based selection strategies also favour examples distant from the current labelled set with the aim of sampling wider, potentially more interesting areas of the feature space. These approaches do not necessarily use a classifier in active learning selection.

Random sampling is one straightforward approach to exploration based selection although it suffers from the fact that it is not well focused and can easily present uninformative examples to the oracle for labelling. Another approach is to use clustering such as the active learning scheme based on the cluster structure of data, presented by Dasgupta & Hsu (2008).

The *Kernel Farthest First* (KFF) algorithm (Baram *et al.*, 2004) was the first widely used exploration based selection strategy. KFF is based on farthest-first traversal sequences in kernel space and iteratively selects the examples that are located farthest away from the currently labelled set for querying. As an exploration based selection strategy, KFF is a little better than random sampling since it tried to select the most dissimilar (or diverse) examples in a dataset.

Density based selection strategy (Donmez *et al.*, 2007) which uses only the density information for active data selection greedily chooses examples that optimise density locally. This strategy can be viewed as one exploration based approach. However, it can be a myopic approach (see Section 6.2.3 and Hu *et al.*, 2010a).

A simple and effective exploration only selection strategy for active learning, namely *Exploration Guided Active Learning* (EGAL) (Section 6.2.3 and Hu *et al.*, 2010a) incorporated diversity with density sampling. It looks for examples that are far away from those already labelled ones and meanwhile in the densest region. EGAL is based only on features of the dataset derived from simple similarity based measures of density and diversity without using the output of a classifier in its selection decisions.

The advantage of exploration based selection strategies is that they explore the entire feature space avoiding the drawbacks associated with exclusively exploitative selection strategies. However, exploration based strategies concentrate on exploration and tend not to improve the learning model very efficiently.

3.3.3 Balancing Exploration and Exploitation

A number of strategies that attempt to balance the exploitation of examples that are near the current decision boundary and the exploration of examples that are representative and/or far from the already-labelled examples have been proposed. One technique applied frequently is to use clustering with exploitation-based selection strategies. Approaches that use clustering tend to talk about the *most representative* example (Tang *et al.*, 2002; Xu *et al.*, 2003), which could either use a local inter-cluster measure which could be considered a density approach, or a global intra-cluster measure which could be considered a diversity approach. For clarity we will first discuss methods using clustering in active learning, and the remainder of this section will discuss techniques under the distinctions of methods using density, methods of using diversity and methods of using both density and diversity.

3.3.3.1 Using Clustering in Active Learning

One approach to balancing exploration and exploitation uses a combination of clustering and uncertainty sampling. The success of using clustering for exploration in active learning is tied to how well the cluster structure aligns with the actual labels (Dasgupta & Hsu, 2008).

Clustering can help in two ways. Firstly, the clustering algorithm is used to calculate a measure to quantify an example's representativeness. For example, Tang *et al.* (2002) used a k-Means clustering algorithm to cluster examples first and calculate the density of each example to quantify its '*representativeness*' inside its cluster which is combined with a measure of '*usefulness*' (based on uncertainty sampling) in a density-weighted scheme to select the examples to present for labelling. Secondly, clustering is used to either explore the whole data space (Cebron & Berthold, 2006; Wiratunga *et al.*, 2003) or the set of examples selected using exploitation based strategies (Xu *et al.*, 2003). Wiratunga *et al.* (2003) introduced a Cluster Utility Score ClUS which combined the average neighbourhood distance and the entropy of each cluster's subset of labelled examples. Then the cluster utility score was used to select the most informative cluster with highest ClUS and then examples to label were selected from the most informative cluster. Cebron & Berthold (2006) used an extended version of fuzzy c-means clustering with noise detection to cluster the data space initially and, after refining the clustering with *Learning Vector Quantisation* (LVQ) , to choose examples at cluster boundaries for labelling. Xu *et al.* (2003) proposed representative sampling which explores the clustering structure for text classification. A k-Means clustering algorithm is employed to cluster examples inside the SVM margin. The medoid documents which are nearest to each cluster center are selected to label.

Nguyen & Smeulders (2004) proposed a framework to incorporate clustering into active learning. In their algorithm (named DWUS by Donmez *et al.* (2007)) a classifier is constructed on the set of centers of clusters, and then the classification decision is propagated to the other examples via a local noise model. The k-Medoids clustering algorithm is used to execute the clustering. Donmez *et al.* (2007) proposed the DUAL algorithm which aims to improve the DWUS algorithm. DUAL is a dynamic approach which adaptively combines a density weighted uncertainty sampling and standard uncertainty sampling without considering density. First DUAL executes DWUS and then switches to use a mixture model for active learning selection at a proper cross-over point by predicting a low derivative of the expected error.

3.3.3.2 Using Density in Active Learning

Incorporating density information with uncertainty sampling has been shown to boost the performance of active learning in various studies (Fujii *et al.*, 1998; McCallum & Nigam, 1998b; Settles & Craven, 2008; Zhang *et al.*, 2002; Zhu *et al.*, 2008b). Fujii *et al.* (1998) applied uncertainty sampling for word sense disambiguation with example-based approach (nearest neighbour approach) in a pool-based setting. An example-based approach is used to determine the correct meaning for a polysemous word. The selective sampling is used to choose the most informative examples (for example sentences) to be labelled with the correct meaning of the specific word in the sentence by the human expert and added to the labelled dataset. Fujii *et al.* (1998) used the neighbours of example x to quantify the increase in the utility score (called *training utility*) of the remaining unlabelled examples if a label is provided for x. The example which is expected to result in the greatest increase in training utility is selected for labelling.

Labelling an example from a highly dense region of the domain space can increase the confidence of the classifications of the neighbourhood. Nearest neighbour information is frequently used in density-based uncertainty sampling. The density of an example is generally calculated as the average similarity of those neighbours of the example within a specified neighbourhood and has been used, for example, to avoid the selection of outliers (Zhu *et al.*, 2008b) and to select the most uncertain examples with maximum density (Zhu *et al.*, 2009). A common approach is to use *density-weighting* where density is defined explicitly and combined as a function of the uncertainty score (Settles & Craven, 2008; Zhu *et al.*, 2008b). For example, an active learning method named *Sampling by Uncertainty and Density* (SUD) was proposed in Zhu *et al.* (2008b) in which a k-Nearest Neighbour based density measure (KNN-density) was adopted to determine whether an unlabelled example is an outlier. The KNN-density for an example x is evaluated by the average cosine similarity (Baeza-Yates & Ribeiro-Neto, 1999) among its K most similar examples:

$$DS(x) = \frac{\sum_{s_i \in S(x)} cos(x, s_i)}{K} \tag{3.2}$$

where $S(x) = \{s_1, s_2, \ldots, s_K\}$ is the set of K most similar examples (nearest neighbours) of x.

3.3.3.3 Using Diversity in Active Learning

Diversity is used in active learning selection strategies mainly in an attempt to overcome the lack of exploration when uncertainty sampling is used. A number of methods for selecting the most informative examples to label, with the consideration of diversity have been described in literature. Diversity has been used to construct diverse labelled sets, diverse batches of examples and diverse committee members.

As suggested in Melville & Mooney (2004), uncertainty sampling "*requires a learner that accurately estimates the uncertainty of its decisions, and tends to oversample the boundaries of it current incomplete hypothesis*". A popular approach to incorporating diversity is to include the *Kernel Farthest First* (KFF) algorithm (which selects those examples that are furthest from the current labelled set) as a member of an ensemble of active learning processes (Baram *et al.*, 2004; Osugi *et al.*, 2005) (the other members of the ensemble are typically based on uncertainty sampling). Baram *et al.* (2004) presented an online algorithm that effectively combines an ensemble of active learners selects the one that currently performs the best to execute. Two top performing active-learning algorithms (SIMPLE MARGIN (Tong & Koller, 2001) and SELF-CONF (Roy & McCallum, 2001)) and a heuristic "Kernel Farthest First" are combined. Osugi *et al.* (2005) proposed an active learning algorithm that balances exploration and exploitation by using KFF to explore and SIMPLE MARGIN to exploit using an SVM classifier. The proposed algorithm dynamically changes between exploration and exploitation with a probability p. This method is in the same spirit as the online algorithm in Baram *et al.* (2004) but in a much simpler fashion.

In the information retrieval literature, several active learning heuristics which capture the diversity of feedback documents have been proposed (Shen & Zhai, 2005; Xu & Akella, 2008b). It has been demonstrated in Shen & Zhai (2005) that the performance of traditional relevance feedback (presenting the top k documents according to relevance only) is consistently worse than that of presenting documents with more diversity. Several practical algorithms based on the diversity of the feedback documents have been presented – for example clustering the documents and choosing the cluster centroids to present for labelling (Shen & Zhai, 2005).

Diversity has been used to avoid repetition among the examples in a batch. Brinker (2003) proposed a diversity-based sampling approach which explicitly incorporates diversity among examples selected in a batch for a batch-mode active learning with SVMs. This approach, namely angle-diversity, considers the overlap in informativeness among examples selected in a batch. The main idea is to select a batch of examples near to the hyperplane, and at the same time, maintain their diversity. A trade off factor λ is introduced to balance between the influence of two components: the distance to the classification hyperplane and the diversity of angles among examples.

3.3.3.4 Using Density & Diversity in Active Learning

Obtaining the label of an example with high density has the advantage that it will affect more unlabelled examples and will increase the classification confidence faster. However, density-based sampling promotes redundancy in the labelled set. The idea of combining diversity with density is to offset this effect. Several active learning algorithms are proposed in the literature that either explicitly (Cebron & Berthold, 2008; Donmez & Carbonell, 2008a; Shen *et al.*, 2004; Wang *et al.*, 2007; Xu *et al.*, 2007) or implicitly (Xu & Akella, 2008a) combine both density and diversity with uncertainty sampling to select examples for labelling. These ensemble-based approaches have proven to be particularly successful as they have the advantages of all three approaches.

There are different ways to incorporate density, diversity to exploitation based selection strategies. For example, Shen *et al.* (2004) proposed two multi-criteria active learning strategies based on informativeness, representativeness and diversity for named entity recognition. The first strategy first chooses "N-best" examples based on informativeness and then clusters the N examples and selects the centroid of each cluster for labelling. The second strategy first combines informativeness and density of a named entity NE using the function $\lambda Info(NE_i)+(1-\lambda)Density(NE_i)$.

Then, a threshold β is used to avoid selecting too similar examples (only examples with $diversity \geq \beta$ will be considered as candidates to label). The second strategy is more efficient than the first one but how to determine the optimal parameters λ and β in real-world applications is a problem.

It has been widely suggested that the relationship between exploitation and exploration should change as the active learning process progresses: at the early stage, exploration is more important so that the wider feature space can be explored. As more examples are labelled and added to the labelled set, exploitation becomes the main concern since the refinement of the classification boundary helps to find the optimal model efficiently (Cebron & Berthold, 2008; Dong & Bhanu, 2003; Donmez *et al.*, 2009). For example, in the self-controlled exploration/exploitation strategy proposed by Cebron & Berthold (2008), during each iteration, the weight of the exploration (finding representative examples in the dataset that are useful to label) decreases whereas the weight of exploitation (adapting the classification boundaries) increases.

3.4 Stopping Criterion

A stopping criterion is used to decide when to stop the active learning process. In most cases a simple stopping criterion that allows the oracle to only provide a specified number of labels, a *label budget*, is used (Novak *et al.*, 2006). Other approaches, referred to as *hold-out accuracy approaches*, stop when the performance of the classifier being built reaches some target performance on a hold-out test set (Campbell *et al.*, 2000). However, stopping criteria such as those which use the characteristics of the classifier (Ertekin *et al.*, 2007; Schohn & Cohn, 2000) are preferable due to the difficulty of getting labelled examples because they do not require a hold-out test set. Schohn & Cohn (2000) suggested stopping when the SVM margin is exhausted, i.e. when there are no unlabelled examples that are closer to the separating hyperplane than any of the support vectors. Researchers have also proposed confidence based stopping criteria which suggest to stop the active learning process based on measuring the classifier confidence (Laws & Schütze, 2008; Vlachos, 2008; Zhu *et al.*, 2010).

3.5 Evaluation Measures

There are a number of different quantitative approaches to evaluating active learning approaches. In classification settings fully labelled datasets are usually used to evaluate the performance of active learning approaches and the performance may be measured in terms of the generalization accuracy (or error) (Schein & Ungar., 2005), F-measure (Ando & Zhang, 2005) or precision-recall break-even point (Tong & Koller, 2001) of the active learner.

Learning curves (for an example see Figure 3.2) are a widely used means to monitor the progress of the labelling process in terms of the 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*, e.g., precision-recall break-even point (Tong & Koller, 2001) and accuracy (Roy & McCallum, 2001; Schohn & Cohn, 2000) on a hold-out test set. From the learning curve a score namely *Area Under the Learning Curve* (AULC) score can be calculated. Based on the AULC

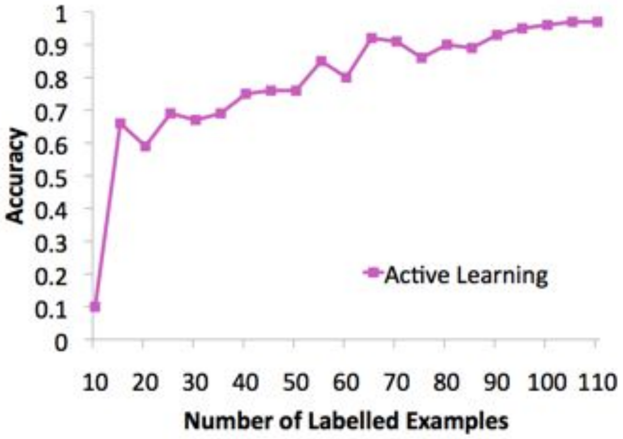


Figure 3.2: A learning curve.

score, more sophisticated measurements can be defined. Famous approaches based on AULC scores include *deficiency* (Baram *et al.*, 2004) and *efficiency* (Raghavan *et al.*, 2006).

In order to measure the efficiency of active learning algorithms, time performance has been used. Hoi *et al.* (2009a) 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.

Since collecting labelled examples is difficult, instead of using a labelled hold-out test set, Schütze *et al.* (2006) proposed several methods for estimating the performance of a classifier from unlabelled data and discussed accuracy estimation for active learning.

3.6 Visualisation and Active Learning

Active learning can facilitate data labelling which is inherently an interactive process. The interactive communication process between the active learner and the human expert is very important. However, the process is not intuitive to human experts. Visualisation is an effective way of helping users to recognize and understand information. This section provides a brief overview of work on visualisation with a focus on work using visualisation in active learning.

3.6.1 Visualisation in Machine Learning

Visualisation is a human oriented method which aims to help people get an intuitive understanding of complex data and abstract information. Visualisation is getting more and more attention in the machine learning research field (Anupam *et al.*, 1994; de Oliveira & Levkowitz, 2003; Keim *et al.*, 2010; Simoff *et al.*, 2008).

In text classification, documents are usually represented as high-dimensional data vectors. A variety of techniques for the visualisation of high-dimensional data have been proposed (see de Oliveira & Levkowitz (2003); van der Maaten & Hinton (2008) for good reviews). Important techniques include icon-based displays such as Chernoff faces (Chernoff, 1973), pixel-based techniques (Keim, 2000), techniques that using histographs for visualising (Ren & Watson, 2005) and model-based visualisation techniques (Kontkanen *et al.*, 2000). Dimensionality reduction techniques such as Principal Components Analysis (PCA; Hotelling (1933)) and vector quantization and projection method such as the Self-Organizing Map (SOM) (Kohonen, 2000) are often used in visualising high-dimensional data.

Information visualisation has demonstrated great advantages in multi-dimensional data analysis. iVIBRATE (Chen & Liu, 2006) offered visualisation-guided disk-labelling solutions that are effective in dealing with outliers, irregular clusters, and cluster boundary extension. Thiel *et al.* (2007) presented two visualisation exploration approaches for analyzing the topic shift of a pool of documents over a given period of time. Mckenna & Smyth (2001) proposed CASCADE – a case-based reasoning authoring tool for visualising the competence of an evolving case base and help the application designers to identify redundant cases for deletion and useful new cases for addition. FormuCaseViz (Massie *et al.*, 2004) visualised the immediate neighbourhood and highlighted features that contribute to similarity and to differences to provide explanation of the case-based reasoning retrieval process for tablet formulation.

3.6.2 Using Visualisation in Active Learning

Visualisation can be very useful in active learning. There are several reasons:

1. **Visually representing datasets**: With the help of visualisation techniques, a better understanding of datasets to run an active learning process on can be gained. Furthermore, some active learning applications contain visual components such as image classification and video annotation.

2. **Easing knowledge acquisition from human experts**: Visualisation techniques can make it easier for human experts to label the selected examples, therefore making the labelling process more interesting and more effective.

3. **Visually evaluating goodness of the selection strategies**: The user needs to get a deep insight of the selection strategy in order to decide if the strategy is optimal. There is also a potential to use visualisation techniques to help in designing selection strategies.

4. **Presenting and explaining the active learning process to end users**: Visualisation can be used to provide explanations that why certain selection strategies work and others do not.

However, research is only now beginning to be carried out on how best to use visualisation techniques in active learning. There are few examples of visualisation applied to active learning. To the best of our knowledge, the earliest work of using visulisation techniques in active learning is done by Abramson & Freund (2005) and Turtinen & Pietikäinen (2005). Abramson & Freund used active learning methods for visual object detection and carried out their experiments using the SEmi-automatic VIsuaL LEarning (SEVILLE) system which provides a graphical user interface for interactive labelling of training examples. Abramson & Freund (2005) demonstrated how an iterative procedure which alternates between training and labelling can be used to reduce the work involved in labelling. Turtinen & Pietikäinen (2005) used a visualisation-based method in their learning system where the self-organizing map was applied to represent original high-dimensional data on a low-dimensional grid.

In a later contribution, namely Visalix (Lecerf & Chidlovskii, 2009), a *Visual Active Learning* (VAL) component which combined the active learning with the visualisation was proposed. VAL was used to guide the user through the labelling process. In VAL, unlabelled examples are represented in a 3D uncertainty space and can be selected by the user for labelling. One of the major limitations in its applicability is that the visualisation is based on the certainty with which the examples can be classified. This makes it hard to plug in popular existing selection strategies which use different classifiers as in VAL.

CBTV-AL (*Case Base Topology Viewer for Active Learning*) (Mac Namee *et al.*, 2010) is another interesting system which is designed to visualise the active learning process so that the operation of selection strategies can be better understood. In CBTV-AL, the *spring model* (Eades, 1984) which allows the display of n-dimensional data on a two dimensional plane based on the similarity between examples is used. Two of the most common similarity measurements – normalised Euclidean distance and cosine similarity are used to measure the similarity between examples (Mac Namee & Delany, 2010). CBTV-AL was used to visualise different active learning processes based on different selection strategies: density sampling (Chapter 6, Hu *et al.* (2010a)), diversity sampling (Chapter 6, Hu *et al.* (2010a)), EGAL approach (Chapter 6, Hu *et al.* (2010a)) and typical uncertainty sampling (Section 9.1). The visualisation was done by showing a graph of the dataset, arranged using the spring model, and annotating this graph to display labels given by the oracle, predictions made using a classifier built from the current dataset and measures of density, diversity and uncertainty (Mac Namee *et al.*, 2010). CBTV-AL provided a visual representation of the active learning process allowing the user a fast intuitive understanding and insight of those selection strategies.

3.7 Reusability Problems in Active Learning

Most of the active learning systems select informative examples for labelling using a selection strategy and then the labelled set can be used to train a classifier. In such systems, usually the classifier used in the selection strategy and later for training a classification model on the labelled set is the same, i.e., labelled examples are reused to build the same type of classifier as that used in the selection strategy. However, in a more general system, different types of classifiers might be used for selection and reuse. According to Tomanek (2010), *sample reuse* is described as a scenario where a set of examples selected by active learning using classifier C_i is used to train another classifier C_j with $C_i \neq C_j$. Sample reuse may happen in the scenario where a cheap, efficient classifier is required in active learning selection while the achieved labelled examples need to be reused to train another, more expensive classifier. Another scenario is that when building the labelled dataset using active learning, the classifier to be trained on the labelled dataset is unknown when labelling. It has been suggested that the resusability problem is a key reason for the reluctance to adopt active learning (Tomanek & Olsson, 2009).

It would be preferable for the data labelled by an active learning process to be reusable as training data for more than one specific classifier particularly in dataset labelling tasks. However, the most informative example for one classifier might not be the most informative example for another classifier since different classifiers value examples differently. Consider the SIMPLE MARGIN algorithm (Tong & Koller, 2001) which picks those examples that are closest to the separating hyperplane of the SVM classifier and a k-NN classifier based active learning method (Hu *et al.*, 2008) which tries to pick examples where the distance of neighbours from one class is the same as that from another class. We show later in Section 8.2 that the most informative examples for the SVM classifier do not work for the k-NN classifier and result in poor reusability when the examples selected for labelling by the SVM-based selection strategy are reused by the k-NN classifier. This is known as the *reusability* problem in active learning (Baldridge & Osborne, 2004; Tomanek *et al.*, 2007).

One of the earliest work in reusability (see Lewis & Catlett, 1994) described a heterogeneous approach in which a classifier of one type (a highly efficient probabilistic classifier) selects examples for training a classifier of another type (the C4.5 rule induction program). Experimental results showed that labelled training examples from one classifier can be used to train another different classifier and achieve better performance than random sampling. However Baldridge & Osborne (2004) used an ensemble-based active learning method for creating labelled training material for Head-driven Phrase Structure Grammar (HPSG) parse selection and reported contrary results concluding that sample reuse is worse than random sampling. This contradictory finding inspired interest in the reusability problem. Tomanek *et al.* (2007) showed that reusability is feasible to a certain extent and argued that by using a committee-based active learner, the dataset built with one type of classifier can reasonably be reused by another classifier. They also showed that the scenario of reusing examples from the same type classifier yielded the best results which indicates that the resultant labelled set favours the base classifier. Both the work of Baldridge & Osborne (2004) and the work of Tomanek *et al.* (2007) suggested that the relationship between the classifiers used in selection and the classifier to be trained in reuse could be a factor of reusability, i.e., the examples selected by the classifier in active learning are more likely to be better reused by another more closely related classifier.

The most comprehensive work to date on reusability has been done by Tomanek & Morik (2010) who systematically investigated the reusability problem for uncertainty sampling. They addressed the question whether and under which conditions examples selected by active learning using one classifier are well-suited as training data for another classifier. Several hypotheses on reusability characteristics and explanatory factors for reusability on general classification problems as well as the natural language processing subtask of Named Entity Recognition were investigated. The following conclusions were made in Tomanek & Morik (2010):

- R1: Labelled examples obtained by active learning with a particular classifier are generally reusable by another classifier.

- R2: For a particular classifier C_j which 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 outperformed by foreign-selection where $C_i \neq C_j$.

- R3: There are no general patterns in reusability, i.e. examples selected by some type of classifier are particularly better reused by another specific classifier.

- R4: A high degree of model similarity between the classifier used in active learning selection and the classifier in reuse often leads to high reusability while a 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. The similarity of two labelled set is estimated based on the divergence of their distributions which is calculated by the Jensen-Shannon divergence (JSD).

Further discussions about these five findings are in Chapter 8.

A classifier C_i which is used as the base classifier in the active learning process is usually referred to as the *selector*, S_i. There are various terms used to describe the classifier which reuses a labelled training set to build its learning model. It is referred to as the *tester* by Tomanek *et al.* (2007), the *reuser* by Baldridge & Osborne (2004) and the *consumer* by Tomanek & Morik (2010). In this work, the terms *selector* and *consumer* are used as follows:

Selector A selector S is a classifier that is used in an active learning process to select examples for labelling to generate a labelled set $\mathcal{L}_S$.

Consumer A consumer T is a classifier which is trained on a labelled set $\mathcal{L}$ which is the output of an active learning process.

A classifier, C, can either be used as a selector, S, in the active learning process or as a consumer, T, in the sample reuse scenario. We extended the typical active learning framework to have a reuse step, as shown in Figure 3.3.

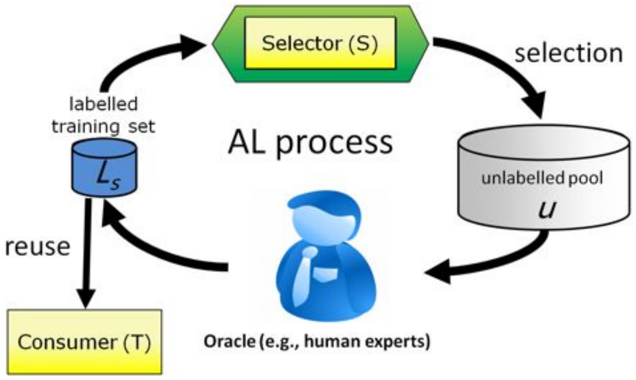


Figure 3.3: Extended pool-based active learning framework.

3.7.1 Reusability Measures

A measure of REUsability (REU) is proposed by Tomanek & Morik (2010) as in Equation 3.3 which 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 x-axis (an interval $[a, b]$).

$$REU(S_{frgn}, S_{self}, S_{base}, a, b) = \frac{AULC(S_{frgn}, a, b) - AULC(S_{base}, a, b)}{AULC(S_{self}, a, b) - AULC(S_{base}, a, b)} - 1 \quad (3.3)$$

where

S_{frgn} is the learning curve of foreign-selection where the classifier used to select examples in active learning is different from the classifier which reuses the labelled examples as training data.

S_{self} is the learning curve of self-selection where the classifier used to select examples is the same as the classifier which reuses the labelled examples for training.

S_{base} is the learning curve of a baseline method, typically random sampling where no classifier is used for selection but examples are selected randomly.

The interpretation of the REU score is that it measures the percentage decrease of the active learning self-selection sampling efficiency by active learning foreign-selection relative to the baseline sampling scenario. The purpose of the baseline sampling scenario S_{base} here is used as a normalisation component. In order to make the REU score meaningful, two assumptions need to be true. The first is that self-selection would constitute the upper efficiency bound for foreign-selection. The second is that baseline selection strategy would constitute the lower bound for foreign-selection. However these two assumptions do not always hold in practical applications. There is also a problem in using random sampling as the baseline since it introduces an element of randomness.

3.8 Other Interesting Fields

The trend of active learning can be viewed as a shift from theoretically motivated methods to heuristics which can be applied in a wide variety of settings. Interesting work includes semi-supervised methods, heuristics employing ensemble techniques for online, dynamic scenarios, cost-sensitive active learning and using active learning for multi-class and multi-label classification.

3.8.1 Semi-supervised Methods

In addition to active learning, other ways of using unlabelled examples include Co-Training, clustering and transductive learning. In recent years, the border topic of “semi-supervised” (see Zhu, 2005, for a survey) or “learning with labelled and unlabelled examples” (Seeger, 2001) has been popular. One example is the Co-Training algorithm (Blum & Mitchell, 1998) which assumes that each example has two conditionally independent feature divisions. Then two separate classifiers can be build from the two feature divisions. Examples which can be classified by each classifier with the highest confidence are added into the training set and the process is repeated. Nigam & Ghani (2000) provided an analysis of why Co-Training algorithms work well and applied Co-Training to datasets without natural feature divisions by manufacturing a feature split. For a common dataset whose feature set is hard to split, semi-supervised algorithms such as semi-supervised EM (Nigam *et al.*, 2000) and Co-EM (Nigam & Ghani, 2000) have been proposed. Semi-supervised EM reduced the need for labelled documents by using unlabelled documents and exploiting information about word co-occurrences that is not accessible from the labelled documents alone to increase classification accuracy in which a multinomial Naïve Bayes classifier was originally used in the work of Nigam *et al.* (2000). Lanquillon (2000) extended the framework so it can be used with any type of classifiers. Transductive learning is another way to make use of the unlabelled examples. For example, Joachims (1999) proposed transductive support vector machines which uses the unlabelled examples with the labelled examples to find the optima parameters for the SVM classifier.

Interesting work which tried to combine semi-supervised learning and active learning has been proposed. Krithara *et al.* (2006) combined semi-supervised and active learning using the *Probabilistic Latent Semantic Analysis* model. Muslea *et al.* (2000) proposed Co-Testing and applied it to problems with redundant views. In a later work, they proposed Co-EMT which is also a multi-view algorithm (Muslea *et al.*, 2002). Co-EMT interleaves active and semi-supervised learning by using Co-Testing to select the most informative examples for the semi-supervised Co-EM algorithm. Yu *et al.* (2009) proposed a unified *Global Entropy Reduction Maximization* (GERM) framework for active learning and semi-supervised learning on speech recognition.

3.8.2 Using Ensembles in Active Learning

Ensemble techniques have been successfully applied in active learning to increase the accuracy and stability of classification (Abe & Mamitsuka, 1998; Baram *et al.*, 2004; Donmez *et al.*, 2007; Zhu *et al.*, 2007). Abe & Mamitsuka (1998) proposed two query-by-committee based selection strategies which used boosting (called *query by boosting*) and bagging (called *query by bagging*) respectively. Melville & Mooney (2004) extended the work of query by boosting and query by bagging to encourage diversity among committee members.

Usually, active learning is designed to select examples for labelling with respect to a single learning algorithm or classifier. Reichart *et al.* (2008) proposed a Multi-Task Active Learning (MTAL) paradigm which selects examples for several annotation tasks rather than for a single one. Sugiyama & Rubens (2008) proposed an approach called ensemble active learning for solving the problem of selecting data and model at the same time.

3.8.3 Cost-sensitive Active Learning

Cost-sensitive active learning approaches have been proposed in the literature to deal with the problem that the cost of acquiring labelled data can vary from one example to the other. For instance in text classification, it might be relatively easy for an oracle to label a shorter document and it might be more time-consuming to label a longer document. Baldridge & Osborne (2004) suggested that there is a cost associated with creating the model itself and this should be factored into the total cost.

Cost-sensitive active learning approaches are discussed more often in *Natural Language Processing* (NLP) tasks (see Culotta & McCallum (2005); Haertel *et al.* (2008); Settles *et al.* (2008)) than in text classification since there are more factors affect the cost of labelling in NLP. Kapoor *et al.* (2007b) described an active learning framework which uses expected *Value-Of-Information* (VOI) to explicitly consider both labelling costs and estimated misclassification costs. Dimitrakakis *et al.* (2008) proposed an approach to deal with labelling cost directly by defining the learning goal as the minimisation of a cost which is a function of the expected model performance and the total cost of the labels used. Donmez & Carbonell (2008b) proposed *proactive learning* which is a generalization of active learning designed to relax unrealistic assumptions and to jointly select the optimal oracle and instance. Proactive learning enables active learning to work under circumstances where cost in label elicitation might be variable and non-uniform. The cost of each example in proactive learning is modeled as a function of the posterior class distribution.

3.8.4 Using Active Learning for Multi-class and Multi-label Classification

Several methods have been proposed for multi-class classification. Yan *et al.* (2003) proposed a unified multi-class active learning approach for automatically labelling video data. Luo *et al.* (2005) presented an active learning approach for the multi-class SVM classifier on the plankton recognition problem.

There is a lot of work been done on multi-label classification approaches in active learning. Li *et al.* (2004b) proposed a multi-label SVM active learning method which uses the one-versus-all method to combine predictions of multiple binary SVM classifiers to solve multi-label image classification problem. Singh *et al.* (2008) addressed a system of using active learning for multi-label image annotation which uses SVM classifiers as the component classifiers. Brinker (2006) proposed a generalisation of pool-based active learning to reduce the labelling effort based on the one-versus-all technique for representing multi-label classifications. Hua & Qi (2008) proposed an online multi-label active learning approach for large-scale video annotation. Esuli & Sebastiani (2009) examined a number of realistic strategies for tackling active learning for multi-label classification by different combination strategies for global labelling where a unique ranking is generated, based on the combination of m confidence scores associated to the same document by m binary classifiers. Other work can be found in Ayache & Quénot (2007a) and Ayache & Quénot (2007b).

3.9 Key Findings

After a detailed literature review on previous work, this section will discuss our key findings.

In previous work, active learning has been mostly used to create high performance classification systems with a limited number of labelled examples. However, active learning can also be very helpful in labelling datasets. We are particularly interested in using active learning to create large collections of labelled examples from unlabelled collections.

Uncertainty sampling is one of the most commonly used selection strategies. Typically, the most informative examples are selected through uncertainty sampling based on direct outputs of classifiers. Instead of using the direct output of the k-NN classifier, a confidence-based selection strategy which uses k-NN based confidence measures to measure the confidence of the prediction and chooses the examples with least confidence for labelling would be better.

Most of the selection strategies use exploitation techniques. Other approaches tend to balance exploitation with exploration. Seldom has work been done on exploration based selection strategies. It is valuable to do research on an exploration only selection strategy which does not need any classifiers, using the idea of exploration.

Active learning is an interactive process which requires the interaction with the oracle. However, it can be hard for human experts to get a deep insight of the selection strategy. Visualisation techniques can be very useful to visualise the active learning process. However seldom work has demonstrated how to use visualise techniques to help the understanding of the active learning process which is what we tend to do.

In the active learning process, a small labelled set is needed to seed the active learning process. In most of the active learning applications, the initial training examples are randomly selected. Previous work has shown that better performance can be achieved by selecting the initial training set using clustering techniques. However, the clustering techniques used are non-deterministic which might be not as good as advanced deterministic clustering techniques. Better and more reliable performance could be achieved by using deterministic clustering algorithms to construct the initial training set.

In a wider domain, there is a reusability problem in using active learning for labelling. It would be useful to compare reusability of popular active learning methods for text classification and identify the best classifier to be used as the selector in the active learning selection strategy and the best classifier to be used as the consumer.

3.10 Conclusions

In this chapter, we reviewed previous research on active learning. Compared to passive learning, active learning is machine learning technique which can be use to select examples for manual labelling in a more informative way instead of randomly picking examples for labelling. Numerous applications have demonstrated its role as a useful technology. Three main forms of active learning are (i) active learning with membership queries, (ii) stream based active learning, and (iii) pool-based active learning. Among them, pool-based active learning has been widely used in text classification which is the one considered in this thesis.

Active learning is typically used to build high performance classifiers. We focus on using active learning to generate labelled datasets. Related work on three major problems while using active learning are discussed which includes how to select an initial training set to seed the active learning process; how to identify the most informative examples to query true labels from the oracle and when to stop the active learning process.

Selection strategies are used to determine how to select the most informative examples. We categorised selection strategies into three groups: exploitation based selection strategies, exploration based selection strategies and selection strategies balancing exploration and exploitation. Exploitation based selection strategies concentrate on examples closest to the classification boundary and thus can refine the classification model efficiently. However, it can be difficult to estimate the classification boundary at the early learning stage with very few initial training examples. Exploration based selection strategies are interested in dense or diverse examples so that they can explore the entire feature space. Selection strategies balancing exploration and exploitation select examples by considering multi criteria, such as uncertainty, density and diversity and have shown superior performance over other selection strategies considering exploitation or exploration alone. Learning curves are often used to evaluate the performance of active learning systems.

This chapter also discusses existing research in using visualisation techniques in active learning and resuability problems in active learning. Visualisation techniques can be used to help human experts in labelling and help users to understand the complex active learning process. In order to build a labelled dataset which can be used to train different types of classifiers, reusability should be considered in active learning.

Other interesting fields related to active learning research including semi-supervised methods, ensembles, cost-sensitive active learning and methods for multi-class and multi-label problems are discussed before we included our key findings and the gaps we are going to fill at the end of this chapter.

The next chapter presents the design of our active learning based labelling system. A preliminary validation of the framework on a recipe dataset shows how this system helps in generating labelled datasets. Experimental methodology discusses the datasets and evaluation measures to be used in this work.

Chapter 4

System Design

This chapter elaborates on the design of our *Active Learning based Labelling* system (ALL) (Hu *et al.*, 2008) which uses ideas from pool-based active learning to investigate the use of active learning approaches in labelling large collections of textual examples. The dual goals of the system are the creation of high-quality labelled datasets and the minimisation of the manual labelling effort. This chapter starts with a description of the design of the system framework. It continues with the discussion of the datasets involved in further experiments and experimental methodologies including the evaluation measures.

4.1 Framework Design of ALL

As discussed in Section 3.1, active learning has been widely used to create classification systems in the absence of large numbers of labelled examples. However, active learning can also be used to create large collections of labelled examples from unlabelled collections. We feel that before a classification system will be valuable, advancements need to be made on the examples labelling problems. Therefore, we focus our efforts on building high quality labelled training datasets which we feel are key sub-problems of the text classification systems. These collections can then be used for disparate purposes beyond classification. ALL is used to generate labelled textual datasets.

The core mechanism of the ALL system is based on pool-based active learning as discussed in Section 3.1.1. A flow diagram of ALL is shown in Figure 4.1. The system starts with a large pool of unlabelled examples, from which a small number of examples are selected and manually labelled by an oracle as the training set. This initial training set can be selected by random sampling from the unlabelled pool or by some advanced methods which will be described in Chapter 7. Given an initial training set, a selection strategy can be built with or without using some classifiers. The selection strategy is used to assign a ranking score as an informativeness measure to each of the unlabelled examples in the pool. Then the most informative example is selected and presented to the oracle for labelling. After the example is labelled it is added into the labelled set and the pool is re-ranked. In this framework, the batch size is set to one so at each active learning iteration only one example from the unlabelled pool is selected for labelling and its label is applied. The process of selecting examples from the pool and re-ranking the pool continues until a label budget is exhausted.

In each active learning iteration, only one example is selected for labelling. In

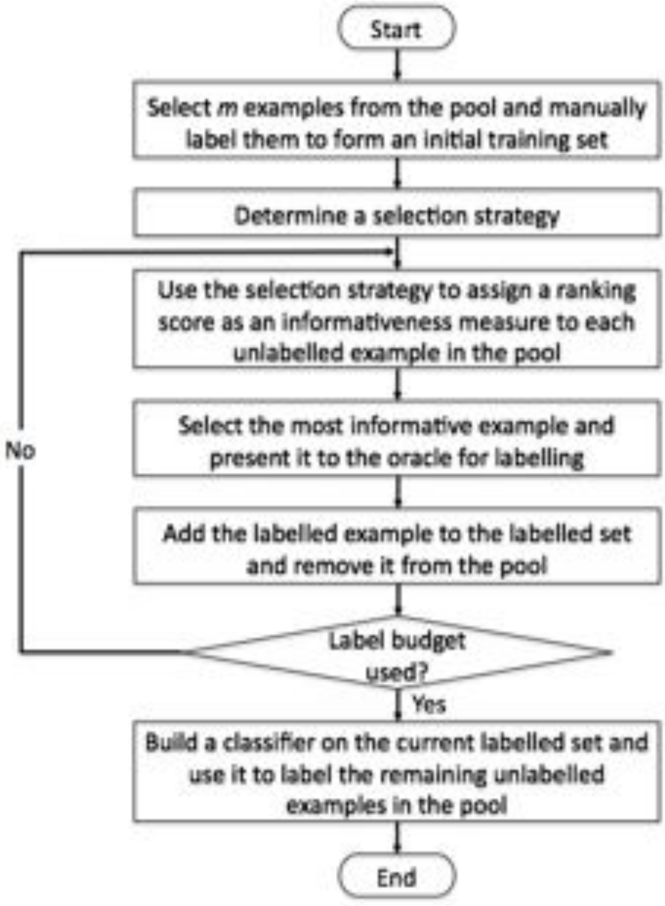

Start
Select *m* examples from the pool and manually label them to form an initial training set
Determine a selection strategy
Use the selection strategy to assign a ranking score as an informativeness measure to each unlabelled example in the pool
Select the most informative example and present it to the oracle for labelling
Add the labelled example to the labelled set and remove it from the pool
Label budget used?
No
Yes
Build a classifier on the current labelled set and use it to label the remaining unlabelled examples in the pool
End

Figure 4.1: The flow diagram of the ALL system.

order to monitor the performance of the proposed system, and compare it to other approaches, after each labelling a classifier is built from the current labelled set, $\mathcal{L}$, and classifications are made for every example remaining in the unlabelled pool, $\mathcal{U}$. These classifications are compared with the actual labels in each dataset and the accuracy of this labelling is recorded. Accuracy is calculated as $Accuracy = C/N$ where N is the size of the union of labelled and unlabelled set and C is the number of correctly labelled examples. Both manually and automatically labelled examples are included in this calculation to measure labelling accuracy over the entire collection, and to ensure that the measure remains stable as the process continues.

4.2 Experimental Methodology

As the preliminary experiment on a recipe dataset showing promising results of using ALL in building labelled datasets, more sophisticated selection strategies and more advanced selection of the initial training set are incorporated to the ALL system. More experiments are executed using the updated ALL system and results are reported in the following several chapters. This section outlines the methodology used in the evaluations performed during the exploration of using ALL to label textual datasets. Firstly, the datasets used are described. Secondly, a description of the evaluative experimental process used in this thesis is provided. Finally, the evaluation measures are discussed.

4.2.1 Datasets Used in Experiments

In order to conduct a comprehensive research, we tested various algorithms proposed in this thesis on seven balanced textual datasets: a spam dataset that contains emails of spam and non-spam (Delany *et al.*, 2005b); four binary classification datasets derived from the widely used 20-Newsgroup collection (20news-18828)[1]; one binary classification dataset from the Reuters collection[2] and one from RCV1 (Lewis *et al.*, 2004). From the 20-Newsgroup collection four datasets, *WinXwin* (consisting of articles from *comp.os.ms-windows.misc* and *comp.windows.x*), *Comp* (consisting of articles from *comp.sys.ibm.pc.hardware* and *comp.sys.mac.hardware*), *Talk* (consisting of articles from *talk.religion.misc* and *alt.atheism*) and *Vehicle* (consisting of articles from *rec.autos* and *rec.motorcycles*) were generated since they represent hard classification tasks among different class combinations in 20-Newsgroup collection. 500 documents were selected from two topics (*earn* and *acq*) of the Reuters-21578 collection, to form the *Reuters* dataset. 500 documents from the RCV1 collection make up the *RCV1* dataset which includes 250 documents from each of the *internal market* (g151) and *external relations* (g158) topics. 500 emails including 250 spam emails and 250 non-spam emails make up of the *Spam* dataset.

Each document is tokenised into words based on letters and non-letters which uses the Unicode specification to decide whether a character is a letter. All non-letter characters are assumed to be separators. Then each document is stemmed using Porter Stemming and stopwords are removed using a common English stop-

[1] http://people.csail.mit.edu/jrennie/20Newsgroups/

[2] http://www.daviddlewis.com/resources/testcollections/reuters21578/

Table 4.1: Benchmark datasets.

Dataset	# of Examples	# of Features	Accuracy
WinXwin	496	8557	91.14%
Comp	500	7044	85.56%
Talk	500	9000	93.92%
Vehicle	500	8059	92.96%
Reuters	500	3692	89.56%
RCV1	500	6135	95.36%
Spam	500	18888	96.80%

words list. After these steps of pre-processing, the tokenised words are used as the features. Normalised TF·IDF weighted word frequency vectors are used to represent documents.

Table 4.1 shows the properties of each dataset. In Table 4.1, the column entitled "# of examples" denotes the number of examples in the dataset, and "# of features" denotes the number of features. The column entitled "accuracy" shows the average accuracy achieved in five iterations of 10-fold cross validation using a 5-NN classifier as an indication of the difficulty of each classification problem.

4.2.2 Configuration of the Active Learning Process

For each dataset, an initial training set containing 10 seed examples is selected using an initial training set selection method. We set the label budget to 110 which includes 10 initial labels and 100 during the active learning process. As the datasets used in the evaluations are fully labelled, the labelling process can be simulated without the need for a human oracle.

In the case of using a classifier as in exploitation-based selection strategies as discussed in Section 3.3.1, usually a k-Nearest Neighbour classifier ($k = 5$), which uses distance-weighted voting, is built. Considering that the initial training set is not big, a smaller k is used in the k-NN classifier. Preliminary experiments showed that a 5-NN to be consistently best. The similarity measure we used is the cosine similarity as shown in Equation 2.2. In typical uncertainty sampling using a k-NN classifier, the example selected to label is the example with the highest ranking score as computed using Equation 9.3. Details of more sophisticated selection strategies and the way to define the ranking score will be discussed in later chapters.

4.2.3 Evaluation Measures

Our major hypothesis is that more sophisticated active learning approaches including selection strategies and initial training set selection methods can be developed to improve traditional approaches. So most of our experiments involve comparing the labelling performance using the newly developed methods with that using typical approaches. We use two performance measures: (1) Learning Curves: This is a standard measure used in active learning to measure the labelling speed as discussed in Section 3.5. (2) AULC scores calculated from the learning curves to analysis the performance of compared approaches in more detail.

We record the performance of each method according to its accuracy, that is the ratio of the number of correctly labelled examples including the manually labelled ones, compared to the total number of examples. Using the accuracy recorded after each manual labelling, a learning curve is plotted with the number of labels given by the oracle on the *x-axis* and labelling accuracy on the *y-axis* (for example Figure 4.2a). The advantage of one algorithm can be visually demonstrated from the learning curves, for example Figure 4.2a shows that the learning curve of Algorithm

Table 4.2: AULC scores of active learning and random sampling.

Algorithms	AULC Score
Algorithm I	80.2
Algorithm II	53.6

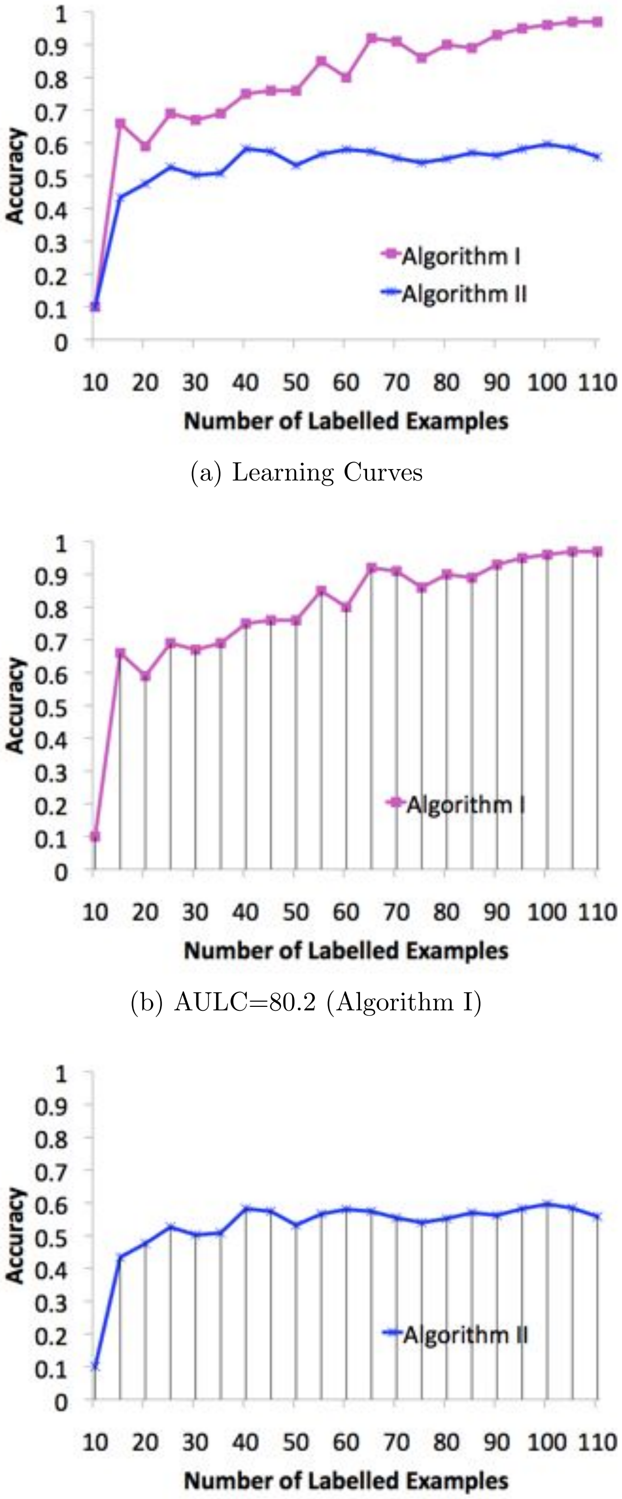
I dominates the learning curve of Algorithm II which indicates that Algorithm I is better than Algorithm II.

As discussed in Section 3.5, AULC scores can be computed from the area under the learning curve. As can be seen from Figure 4.2b and 4.2c, the AULC score of Algorithm I is higher than the AULC score of Algorithm II. So, if the performances were to be ranked based on AULC scores in descending order, Algorithm I would be ranked as the first while Algorithm II being ranked as the second as shown in Table 4.2.

4.3 Conclusions

This chapter outlined the design of ALL, our Active Learning based Labelling system for text classification. The basis of the system is as follows. First, provided with several initially labelled examples, the system utilises some selection strategy to select the most informative examples for labelling by the oracle.

In this chapter we have also outlined the experimental methodology. Seven commonly used datasets in text classification domain are described which will be used in further experiments in later chapters. TF·IDF weighting scheme is used. Configuration of the active learning process is then discussed. 10 examples are selected and manually labelled as initial training examples. The active learning



(a) Learning Curves

(b) AULC=80.2 (Algorithm I)

(c) AULC=53.6 (Algorithm II)

Figure 4.2: Illustrative learning curves and AULC scores calculations.

process is run to a label budget of 110. Learning curves and AULC scores are used to measure the system performance.

Chapter 5

Confidence-based Active Learning Using Aggregated Confidence Measure

Our previous work (see Section 9.1 and Hu *et al.*, 2008) used uncertainty sampling based on a k-NN classifier to label a set of recipes for information retrieval-like search queries. Experimental results showed that reasonably accurate labels can be applied to a large recipe dataset with a human labelling requirement of just 15% of the total number of recipes. Instead of using the direct output of the k-NN classifier, this chapter introduces work done on investigating using k-NN based confidence measures to measure the confidence of the prediction and choose the examples with least confidence for labelling in a new active learning selection strategy, and shows how the performance of this strategy is better than one based on uncertainty sampling using classification scores (Hu *et al.*, 2009).

5.1 Introduction

As discussed in Section 3.3.1.1 and can be seen in Section 9.1 as an example, the typical approach to uncertainty sampling is to use the output of a ranking classifier that produces numeric *classification scores* (e.g. k-Nearest Neighbour, Naïve Bayes or support vector machines) as a measure of *classification confidence*. However, Delany *et al.* (2005a) have shown that there is not a direct relationship between classification scores and classification confidence. This suggests that active learning selection strategies that measure certainty using factors other than classification scores would be more effective. Delany *et al.* (2005a) showed that an aggregate of five basic confidence measures used with k-NN classifiers are particularly effective in estimating classification confidence. In this chapter we investigate an active learning selection strategy based on these confidence measures, and evaluate whether this performs better than a selection strategy based on classification scores.

The remainder of this chapter is organised as follows. Section 5.2 will describe our overall active learning approach including the details of how the confidence measures are integrated into the selection strategy. This *confidence-based selection strategy* has been evaluated against a strategy based on classification scores using a number of text datasets and the results of these evaluations will be presented and discussed in Section 5.3. Finally, we conclude in Section 5.4.

5.2 Approach

This section will describe our approach to confidence-based active learning methods using confidence measures of a k-NN classifier.

5.2.1 Confidence Measures

This section describes three measures which are used in our confidence-based selection strategy.

Confidence measures proposed by Delany *et al.* (2005a) are used in our work. The objective of the k-NN measures is to assign higher confidence to those examples that are 'close' (i.e. with high similarity) to examples of its predicted class, and are 'far' (i.e. low similarity) from examples of a different class. The closer a target example is to examples of a different class, the higher the chance that the target example is lying near or at the decision surface. Whereas the closer an example is to other examples of the same class, the higher the likelihood that it is further from the decision surface. All the k-NN measures used in our confidence-based selection strategy perform some calculation on a ranked list of neighbours of a target example using a combination of:

- the distance between an example and its nearest neighbours ($NN_i(t)$ denotes the ith nearest neighbour of example t),
- the distance between the target example t and its nearest like neighbours ($NLN_i(t)$ denotes the ith nearest *like* neighbour to example t),

- the distance between an example and its nearest unlike neighbours ($NUN_i(t)$ denotes the ith nearest *unlike* neighbour to example t).

Preliminary experiments using the five measures proposed in Delany *et al.* (2005a) showed a high correlation between three of them, and so we chose to use the three of the five that are least correlated in our evaluations. Full details on each measure can be found in Delany *et al.* (2005a). Small changes were made to two of the three selected confidence measures by introducing a smoothing parameter ϵ.

Average NUN Index The *Average Nearest Unlike Neighbour Index* (AvgNUNIndex) is a measure of how close the first k NUNs are to the target example t as given in Equation 5.1.

$$AvgNUNIndex(t,k) = \frac{\sum_{i=1}^{k} \mathit{IndexOfNUN}_i(t)}{k} \tag{5.1}$$

where $\mathit{IndexOfNUN}_i(t)$ is the index of the ith nearest unlike neighbour of target example t, the index being the ordinal ranking of the example in the list of NNs.

Similarity Ratio The *Similarity Ratio* measure (SimRatio) calculates the ratio of the similarity between the target example t and its k NLNs to the similarity between the target example and its k NUNs, as given in Equation 5.2.

$$SimRatio(t,k) = \frac{\sum_{i=1}^{k} Sim(t, NLN_i(t)) + \epsilon}{\sum_{i=1}^{k} Sim(t, NUN_i(t)) + \epsilon} \tag{5.2}$$

where $Sim(a,b)$ is the similarity between examples a and b and ϵ is a smoothing value to allow for situations where an example may have no NLNs or NUNs ($\epsilon = 0.0001$

is used in all of our evaluations). This could happen when only the target example is from one class and all the remaining examples (neighbours) are from the other class.

Similarity Ratio Within K The *Similarity Ratio Within K* (SimRatioK) is similar to the Similarity Ratio as described above except that, rather than consider the first k NLNs and the first k NUNs of a target example t, it uses only the NLNs and NUNs from the first k neighbours. It is defined in Equation 5.3.

$$SimRatioK(t,k) = \frac{\sum_{i=1}^{k} Sim(t, NN_i(t))\delta_{t,NN_i(t)}}{\epsilon + \sum_{i=1}^{k} Sim(t, NN_i(t))(1-\delta_{t,NN_i(t)})} \tag{5.3}$$

where $Sim(a,b)$ is as above, δ_{ab} is Kronecker's delta where $\delta_{ab} = 1$ if the class of a is the same as the class of b and 0 otherwise, and ϵ is a smoothing value to allow for situations where an example may have no NUNs among its k nearest neighbours ($\epsilon = 0.0001$ is used).

5.2.2 Single Confidence Measure Algorithm

Before any of the confidence measures can be used to calculate classification confidence it is necessary to identify for each measure a confidence threshold for each of the possible classes. Predictions with confidence higher than the predicted class's threshold are considered *confident*, while those with confidence below are considered *non-confident*. A confidence threshold consists of two values: the k value indicating the number of neighbours to use in the confidence calculation, and the actual threshold value. The threshold value for a particular class is the value that results in the highest proportion of correctly predicted examples of a particular class when there were no incorrect predictions. The confidence thresholds are referred to as $thres_{ij}$ and k_{ij} for each confidence measure M_i ($i = 1 \dots n$), and each class $j = 1 \dots c$.

The approach used for identifying the thresholds is to perform a leave-one-out validation on each training set (TR) calculating the measure values for a series of k values, from $k = 3$ to $k = |TR| - 1$. Over all the k values, the confidence value which gives us the highest proportion of correctly classified examples with no false positives is used as the confidence threshold. It might result in unreasonable confidence values when the initial training set is too small. Usually the size of the training set ($|TR|$) in confidence computation is bigger than 30. Specific details on the approach used for setting the threshold level for a class are described in Delany *et al.* (2005a).

Our approach to selection using a single confidence measure is as follows. First, the confidence threshold for the measure for each class is identified from the current labelled set as described above. Then each example in the pool is classified using the current labelled set and the value of the confidence measure is calculated and compared to the confidence threshold value. Examples with confidence values higher than the confidence threshold are added to the *confident set*. Otherwise, they are put into the *non-confident set*. The example in the non-confident set with the smallest confidence value is selected to present to the oracle for labelling first. If the non-confident set is empty, the example selected is the one in the confident set with the smallest confidence value. The pseudo-code for the confidence measure algorithm is presented in Algorithm 2. In Algorithm 2, two lists of confidence values are maintained. One is for the examples classified with confidence and the other for those classified without confidence. It can not be simplified by using only one ranked list whereby candidates are ranked in ascending order of confidence because the confidence thresholds for examples predicted as positive and negative might be different and an example with higher confidence value is not necessarily the one being classified with confidence.

```
Input: An initial labelled training set $\mathcal{L}$, an unlabelled pool $\mathcal{U}$, a $k$-NN
       classifier $\mathcal{C}$, a stopping criterion $\mathcal{SC}$, a confidence measure $CM$ and a
       batch size $b$
Output: A labelled dataset
while The stopping criterion $\mathcal{SC}$ is not met do
    Identify the threshold of the confidence measure $CM$, find the $thres_j$, $k_j$,
    for every class $j$;
    $ConfSet = \emptyset$, $NonConfSet = \emptyset$, $Selected = \emptyset$;
    foreach Example $e$ in the pool $\mathcal{U}$ do
        Classify $e$ using the classifier $\mathcal{C}$;
        Calculate confidence value $m$ using $k_j$ for $j$ = predicted class of $e$;
        if the confidence value $m < thres_j$ then
            $NonConfSet = NonConfSet \cup \{e\}$;
            Set the confidence score of $e$ $conf(e) = m$;
        else
            $ConfSet = ConfSet \cup \{e\}$;
            Set the confidence score of $e$ $conf(e) = m$;
        end
    end
    foreach $l, l = 1 \ldots b$ do
        if $NonConfSet == \emptyset$ then
            $Selected = Selected \cup \{e_l\}$ where
            $conf(e_l) = min(conf(e)), e_l \in ConfSet$;
            $ConfSet = ConfSet/\{e_l\}$;
        else
            $Selected = Selected \cup \{e_l\}$ where
            $conf(e_l) = min(conf(e)), e_l \in NonConfSet$;
            $NonConfSet = NonConfSet/\{e_l\}$;
        end
    end
    Label each $e_l \in Selected$ ;
    $\mathcal{L} = \mathcal{L} \cup Selected$ , $\mathcal{U} = \mathcal{U}/Selected$ ;
end
```

Algorithm 2: The single confidence measure algorithm

5.2.3 Aggregated Confidence Measures Selection Strategy

Our *Aggregated Confidence Measures Selection strategy* (ACMS) aggregates the three confidence measures into a new selection strategy. First, a number of c thresholds are identified for each individual measure for c classes. Each example e in the pool is classified using the initial training set and the value for each confidence measure m_i is calculated. Based on the predicted class of the example the appropriate threshold value is checked for each of the measures. Preliminary experimental results confirmed the suggestion of Delany *et al.* that a liberal aggregation strategy works better than a conservative one. So if any one of the measures indicates confidence, i.e. $m_i > thres_{ij}$ for any $i = 1 \ldots n$ and $j = \textit{the predicted class}$, then we consider that the example has been classified with confidence, and it gets added to the *confident set*. Otherwise, it gets added to the *non-confident set*.

Different strategies for combining confidence measures were considered in a range of preliminary experiments which showed the min/max combination to be consistently best. The min/max combination works as following: for an example e classified with confidence, a confidence value $conf(e)$ is assigned the value that indicates most confidence, i.e. $conf(e) = max(m_i)$ for those M_i's that indicate confidence; while the one used for an example in the non-confident set should be the m_i that indicates least confidence (i.e. $conf(e) = min(m_i)$ for those M_i's that do not indicate confidence).

In order to be able to compare m_i across different confidence measures, the values of m_i for each confidence measure M_i are normalised using statistical normalisation after performing a log transformation to correct those with skewed distributions.

Once all pool examples have been classified, the one that the classifier is least confident of is the example in the non-confident set that has the smallest $conf(e)$ value. If the non-confident set is empty, the least confident example is the one in the confident set with the smallest $conf(e)$ value. This is the example that is presented to the oracle for labelling before the process repeats until the stopping criteria is met. The algorithm for our ACMS strategy is presented in Algorithm 3.

5.3 Evaluation

The two objectives to the evaluations described here were to confirm the superiority of using an aggregate confidence measure over using single confidence measures; and to compare the performance of our ACMS approach with an uncertainty sampling approach based on classification scores.

Previous work has shown that using clustering to select the initial training set gives better results than random selection (Kang *et al.*, 2004). However, work used non-deteministic clustering algorithms which can lead to highly inconsistent results over many trials as non-deteministic clustering algorithms are quite unstable, especially when dealing with high dimensional textual data (Hu *et al.* (2010c), further details on this will be discussed in Chapter 7). For this reason, we use a simple Furthest-First-Traversal clustering technique (Greene, 2006) which is deterministic and will always return the same initial training set for a given dataset.

We evaluated the performance of selection strategies using each individual confidence measure and using the aggregation of the measures on all of the seven datasets

```
**Input**: An initial labelled training set $\mathcal{L}$, an unlabelled pool $\mathcal{U}$, a $k$-NN
        classifier $\mathcal{C}$ for classes $1 \ldots c$, a stopping criterion $\mathcal{SC}$, a batch size $b$, a
        set of confidence measures $M_i$, $i = 1 \ldots n$
**Output**: A labelled dataset
while $\mathcal{SC}$ is not met do
    foreach confidence measure $M_i, i = 1 \ldots n$ do
        Identify the threshold: find $thres_{ij}$ and $k_{ij}$, for $j = 1 \ldots c$;
    end
    $ConfSet = \emptyset$, $NonConfSet = \emptyset$, $Selected = \emptyset$;
    foreach example $e \in \mathcal{U}$ do
        Classify $e$ using the classifier $\mathcal{C}$;
        Calculate $m_i$ using $k_{ij}$ for $i = 1 \ldots n$ and $j$ = predicted class of $e$;
        if $m_i > thres_{ij}$ for any $i = 1 \ldots n$ and $j$ = predicted class of $e$ then
            $ConfSet = ConfSet + e$ ;
            Set the confidence score: $conf(e) = max(m_i)$;
        else
            $NonConfSet = NonConfSet + e$;
            Set the confidence score: $conf(e) = min(m_i)$;
        end
    end
    foreach $l, l = 1 \ldots b$ do
        if $NonConfSet == \emptyset$ then
            $Selected = Selected \cup \{e_l\}$ where
            $conf(e_l) = min(conf(e)), e_l \in ConfSet$;
            $ConfSet = ConfSet / \{e_l\}$;
        else
            $Selected = Selected \cup \{e_l\}$ where
            $conf(e_l) = min(conf(e)), e_l \in NonConfSet$;
            $NonConfSet = NonConfSet / \{e_l\}$;
        end
    end
    Label each $e_l \in Selected$ ;
    $\mathcal{L} = \mathcal{L} \cup Selected$ , $\mathcal{U} = \mathcal{U} / Selected$ ;
end
```

Algorithm 3: The algorithm for the Aggregated Confidence Measure Selection strategy

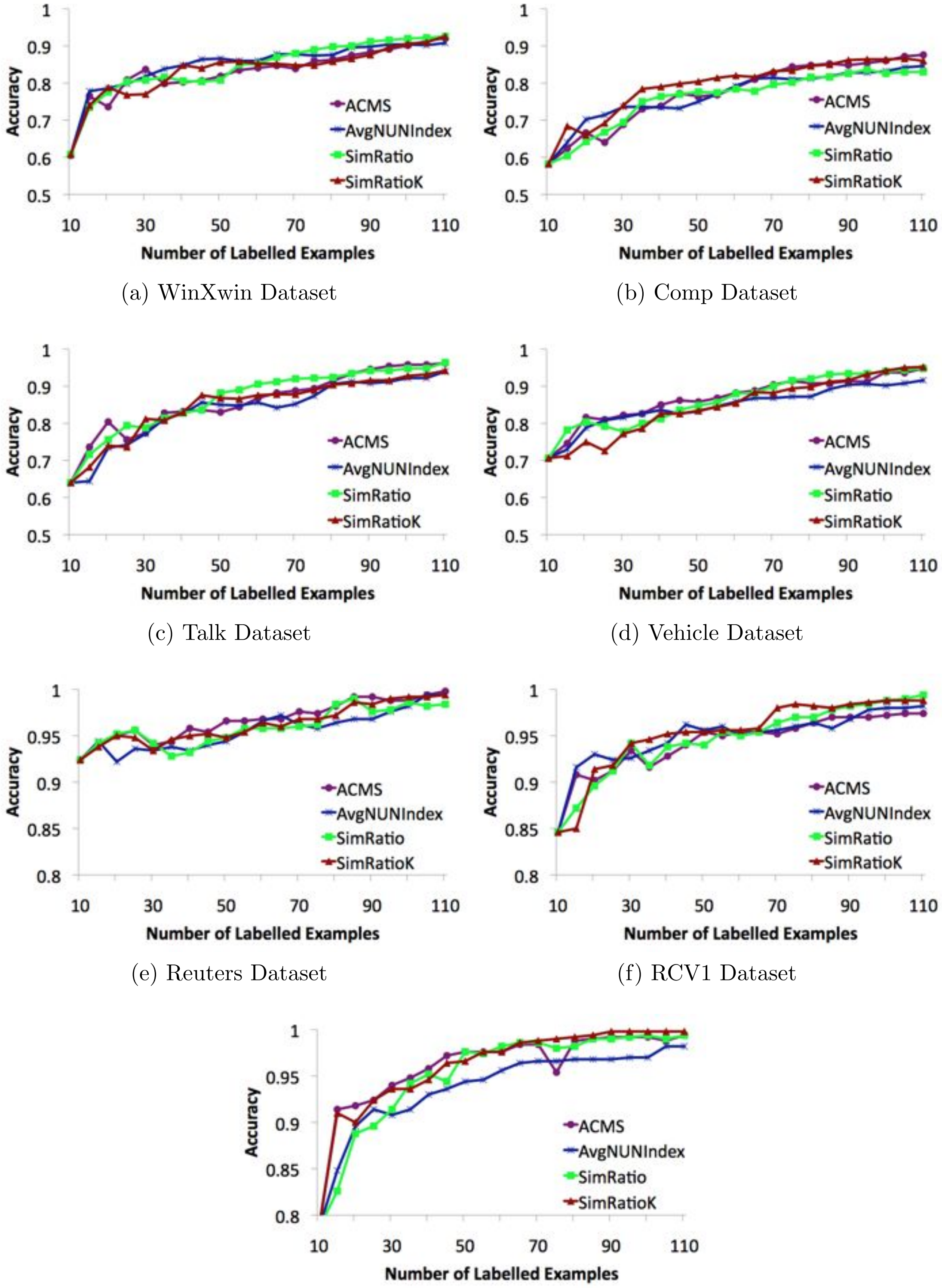
described in Section 4.2.1. Graphs of learning curves on seven datasets are shown in Figure 5.1. The results indicate that the learning curve for ACMS is at least as good as but dominates curves for the individual measures on three out of the seven datasets.

Furthermore, Table 5.1 shows the AULC scores of the four compared methods. For each dataset, the rank of every method is shown in parentheses. The average rank across the ranks over the seven datasets is included too. Look at the learning curves in Figure 5.1, ACMS generates the best performance on three datasets while SimRatioK performs best on two datasets. But look at the ranks in Table 5.1, the two methods appear to have comparable performance, i.e., ACMS's performance is not superior to SimRatioK. One possible reason is that the log transformation might not be the best way to correct those skewed distributions to perform the statistical normalisation.

Table 5.1: AULC scores of the AvgNUNIndex, SimRatio, SimRatioK and ACMS algorithms.

	ACMS	AvgNUNIndex	SimRatio	SimRatioK
WinXwin	83.7(4)	85.2(1)	84.5(2)	83.9(3)
Comp	77.7(2)	77.0(3)	76.4(4)	79.7(1)
Talk	86.2(2)	83.8(4)	87.1(1)	85.3(3)
Vehicle	87.1(1)	84.9(4)	86.5(2)	85.0(3)
Reuters	96.7(1)	95.5(4)	95.9(3)	96.2(2)
RCV1	94.3(4)	95.1(2)	94.8(3)	95.5(1)
Spam	96.5(1)	94.2(4)	95.3(3)	96.4(2)
Avg. Rank	**2.1**	**3.1**	**2.6**	**2.1**

Interestingly, across all ACMS experiments the average *effectiveness* – how often *conf*(e) for an examples is determined by a particular confidence measure – of Avg-NUNIndex, SimRatio and SimRatioK are 38.87% 34.57% and 26.56% respectively as



(a) WinXwin Dataset

(b) Comp Dataset

(c) Talk Dataset

(d) Vehicle Dataset

(e) Reuters Dataset

(f) RCV1 Dataset

(g) Spam Dataset

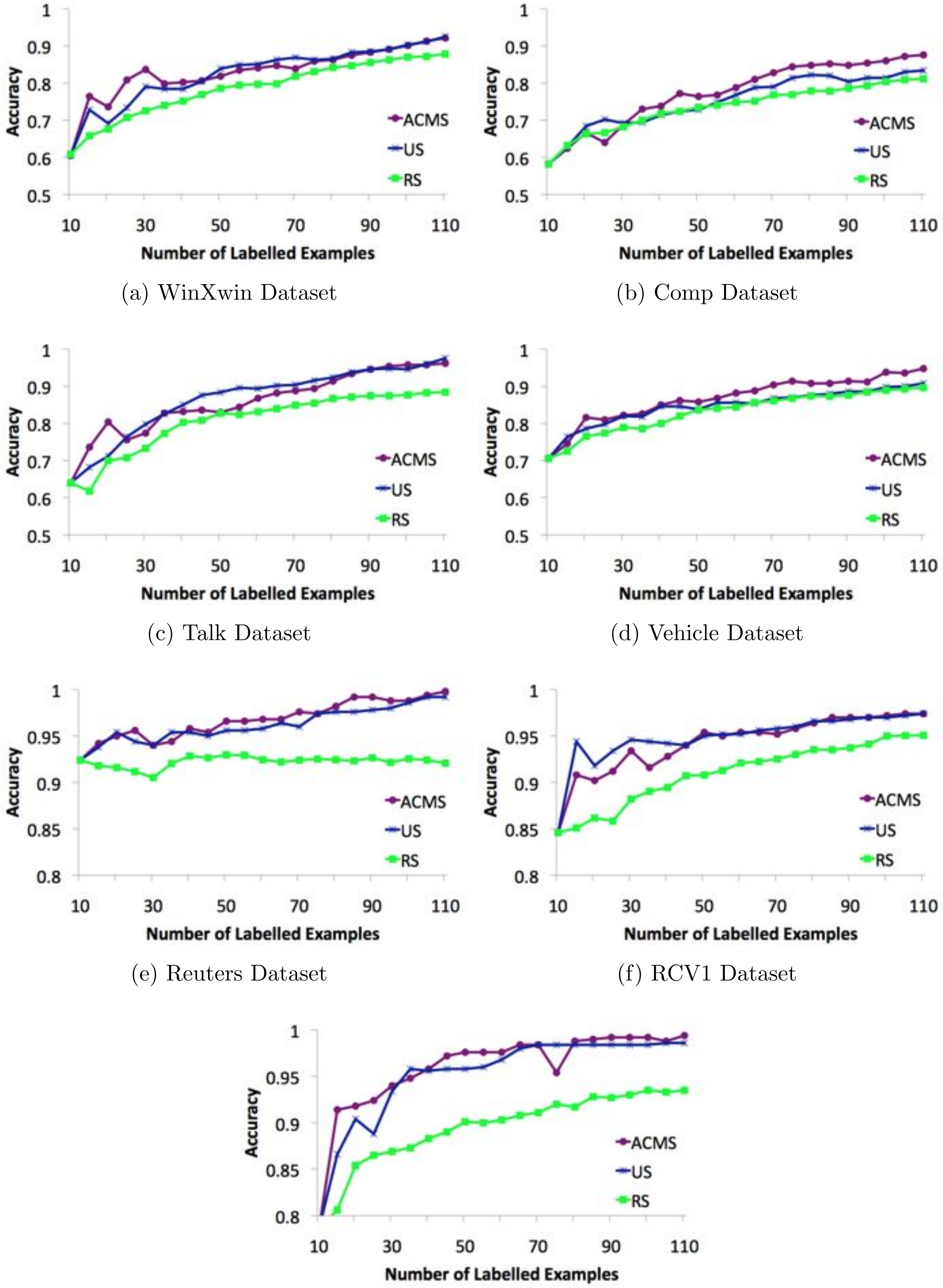
Figure 5.1: Comparison of individual confidence measures and ACMS as the selection strategy. Axes are zoomed for reading.

Table 5.2: Effectiveness of the AvgNUNIndex, SimRatio and SimRatioK confidence measures in the ACMS algorithm.

	AvgNUNIndex	SimRatio	SimRatioK
WinXwin	34.38%	28.85%	36.77%
Comp	39.73%	33.14%	27.13%
Talk	37.46%	40.83%	21.71%
Vehicle	43.81%	34.35%	21.85%
Reuters	38.72%	29.55%	31.74%
RCV1	39.91%	37.86%	22.23%
Spam	38.12%	37.41%	24.47%
Average	38.87%	34.57%	26.56%

shown in Table 5.2 which indicate that three confidence measures contribute almost equally in determining the confidence of the example in the aggregating scheme.

Figure 5.2 shows the results of comparing the ACMS strategy with the more typical Uncertainty Sampling (US) strategy using classification scores. A Random Sampling (RS) strategy, which randomly picks the example to label, is also included as a baseline. The learning curve for the ACMS strategy dominates the curve for the RS strategy in all cases, and the curve for the US strategy on five (WinXwin, Comp, Vehicle, Reuters, Spam) of the seven datasets. Table 5.3 shows the AULC scores of the three compared methods. The performance of both ACMS and US over RS was found to be significant (at $\alpha = 0.05$) using the Wilcoxon signed-rank test. However, since ACMS didn't perform well on Talk and RCV1 dataset, no significant difference between US and ACMS can be found using the Wilcoxon signed-rank test. One plausible reason for the poor performance of ACMS over US on Talk and RCV1 dataset is that the initial training set construction method (FFT) can not work very well with these two datasets.



(a) WinXwin Dataset

(b) Comp Dataset

(c) Talk Dataset

(d) Vehicle Dataset

(e) Reuters Dataset

(f) RCV1 Dataset

(g) Spam Dataset

Figure 5.2: Comparison of the ACMS, US and RS selection strategies. Axes are zoomed for reading.

Table 5.3: AULC scores of the ACMS, US and RS selection strategies.

	ACMS	US	RS
WinXwin	83.7(1)	82.6(2)	78.7(3)
Comp	77.7(1)	75.5(2)	73.8(3)
Talk	86.2(2)	87.1(1)	80.8(3)
Vehicle	87.1(1)	84.9(2)	83.2(3)
Reuters	96.7(1)	96.4(2)	92.3(3)
RCV1	94.3(2)	95.1(1)	91.0(3)
Spam	96.5(1)	95.5(2)	89.5(3)
Avg. Rank	1.3	1.7	3.0

5.4 Conclusions

In this chapter, we propose a new selection strategy for active learning using k-NN based confidence measures. Selection strategies based on three confidence measures and an aggregated confidence measure of the three single measures are developed. Experimental results show that selection strategies based on confidence measures perform well and the selection strategy based on the aggregated confidence measure gives slightly better results than those based on single confidence measures but no significant improvement can be found using the Wilcoxon signed-rank test. We also compared the selection strategy based on the aggregated confidence measure to the more typical uncertainty sampling approach using the direct output of classifiers. Although the difference is not significant, the results indicate that the approach based on the aggregated confidence measure is better than the typical uncertainty sampling method. Possible reasons could be that the size of the initial training set is too small for the calculation of confidence values and the use of the log transformation might not be the best method to use. Furthermore, we have found later (Hu *et al.*, 2010c, Chapter 7) that the Furthest-First-Traversal can performs badly on some datasets.

Chapter 6

EGAL: Exploration Guided Active Learning Selection Strategy

In this chapter, a simple but effective exploration-only selection strategy is described. This approach uses nearest neighbour based density and diversity measures to explore the feature space. We show how its performance is comparable to the more computationally expensive exploitation based approaches and offers the opportunity to be classifier independent.

6.1 Introduction

As discussed in Section 3.3.1, uncertainty sampling is considered an *exploitation*-based active learning selection strategy which attempts to refine the classification decision boundary in uncertain areas of the feature space and can work well if the initial classification boundary is well shaped. However, with small numbers of labelled examples, it can be difficult to reliably estimate the boundary, and it has been suggested that exploitation techniques are prone to querying outliers (Roy & McCallum, 2001). To overcome this problem, selection strategies have been developed which attempt to balance exploitation with *exploration*, focussing on examples distant from the labelled set with the aim of sampling wider, potentially more interesting areas of the feature space (see Section 3.3.3 for more details). These multi-faceted approaches have recently become popular.

However, there is not much work done on exploration-only selection strategy. To the best of our knowledge, no approach has been described in the literature that combines density sampling and diversity sampling without also using uncertainty sampling. We believe that by applying an exploration-only approach to active learning selection we can create an active learning system that is based only on features of the dataset derived from a similarity measure, and does not suffer from the difficulties associated with exploitation-based approaches. Furthermore, using an exploration-only approach is efficient as it does not require the repeated re-training of a classifier and re-classification of the unlabelled dataset associated with exploitation-based approaches.

In this chapter we present *Exploration Guided Active Learning* (EGAL), a simple, computationally efficient, exploration-only active learning selection strategy that does not use the output of a classifier in its selection decisions. We compare the performance of this new approach to existing exploitation-based and hybrid selection strategies on a selection of text classification datasets.

The rest of the chapter is organised as follows: We introduce our exploration-based selection strategy, EGAL, in Section 6.2 showing how it incorporates simple similarity-based measures of density and diversity. Section 6.3 describes an evaluation of EGAL using seven textual datasets. Section 6.4 shows examples of using visualisation techniques in understanding multi selection strategies including EGAL. We conclude in Section 6.5 discussing how this approach can help in active learning.

6.2 The Exploration Guided Active Learning Algorithm

This section describes our exploration-only active learning selection strategy: EGAL. We first discuss how we measure density and diversity, and then explain how they are combined.

6.2.1 Measuring Density

The density of an example can be measured by how many examples are near to it or how dense its surrounding area is which can be quantified by similarities. We measure the density of an unlabelled example x_i by considering the similarity of x_i to the examples that are within a pre-defined neighbourhood N_i of x_i, as given in Equation 6.1. This neighbourhood N_i (see Equation 6.2) is set by a similarity threshold α, where $\alpha = \mu - 0.5 \times \delta$; μ and δ being the mean and standard deviation of the pair-wise similarities of all examples in $\mathcal{D}$ respectively. We set the value of α in this way as preliminary experiments showed that it gives the best results.

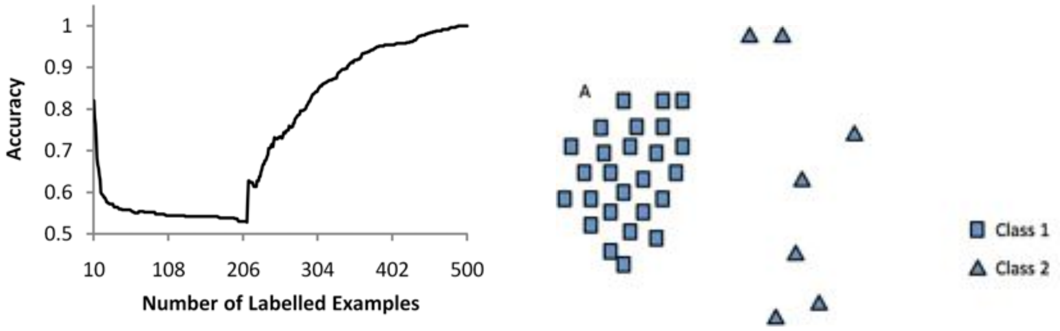
$$
\begin{aligned}
\text{density}(x_i) &= \sum_{x_r \in N_i} \text{sim}(x_i, x_r) && (6.1)\\
N_i &= \{x_r \in \mathcal{D} | \text{sim}(x_i, x_r) \geq \alpha\} && (6.2)
\end{aligned}
$$

The density depends on the whole dataset which includes both the labelled set $\mathcal{L}$ and the unlabelled pool $\mathcal{U}$. Unlike other density measures such as the one in He & Carbonell (2007), we use the sum of the similarities in the neighbourhood N_i instead of the count of the number of neighbours in N_i. The effect of this is to have fewer *ties* in the density-based ranking, which makes for a more straightforward density-based sampling technique. A selection strategy using density alone will select the example(s) with the highest density to present for labelling.

6.2.2 Measuring Diversity

We measure diversity by considering the examples which are most dissimilar to the labelled set $\mathcal{L}$. Distance being the inverse of similarity, our diversity measure for an example x_i (given in Equation 6.3) is defined as the distance between x_i and its nearest labelled neighbour. The diversity measure has the advantage of efficient time complexity and it also ensures that the newly selected examples are different from the examples already in $\mathcal{L}$. A selection strategy based on diversity alone would select the example(s) with highest diversity to present for labelling.

$$
\text{diversity}(x_i) = \frac{1.0}{\max_{x_r \in \mathcal{L}} \text{sim}(x_i, x_r)} \qquad (6.3)
$$



(a) (b)

Figure 6.1: Illustrating how density-based sampling can perform badly.

6.2.3 Combining Density and Diversity

Density and diversity sampling greedily choose examples that optimise locally, which can make them myopic approaches to selection in active learning. They can become trapped in local optimums which can result in poor performance globally. An example of density sampling's poor performance is evident in Figure 6.1a, which shows the performance of a density-based active learner on the Reuters dataset. This shows a degradation in performance until after 200 or so examples are labelled, at which point performance improves rapidly. Figure 6.1b illustrates how this can happen. With density sampling, examples from class 1 in group A will be repeatedly selected for labelling while examples from class 2 will be ignored, leading to a poorly defined classification boundary during this time. When diversity alone is used, similarly dysfunctional scenarios can arise.

To overcome these problems, we introduce an element of diversity to a density-based sampling approach. Including diversity means that high density examples that are close to labelled examples are not selected for labelling by the oracle.

To determine whether an example should be considered as a candidate for selection, we use a threshold β. If the similarity between an unlabelled example x_i and its nearest neighbour in the labelled set is greater than β then x_i is not a candidate for selection. We call the set of examples that can be considered for selection the *Candidate Set*, $\mathcal{CS}$, which we define as in Equation 6.4:

$$\mathcal{CS} \quad = \quad \{\exists x_i \in \mathcal{U} \mid diversity(x_i) \geq \frac{1}{\beta}\} \tag{6.4}$$

Our EGAL selection strategy ranks the possible candidates for selection (i.e. those in $\mathcal{CS}$) based on their density, and selects those examples with the highest density for labelling first. Thus, examples close to each other in the feature space will not be selected successively for labelling.

Parameters α and β play an important role in the selection process. α controls the radius of the neighborhood used in the estimation of density, while β controls the radius of the neighbourhood used in the estimation of $\mathcal{CS}$. The values selected for these parameters can significantly impact the overall performance, especially the value of β.

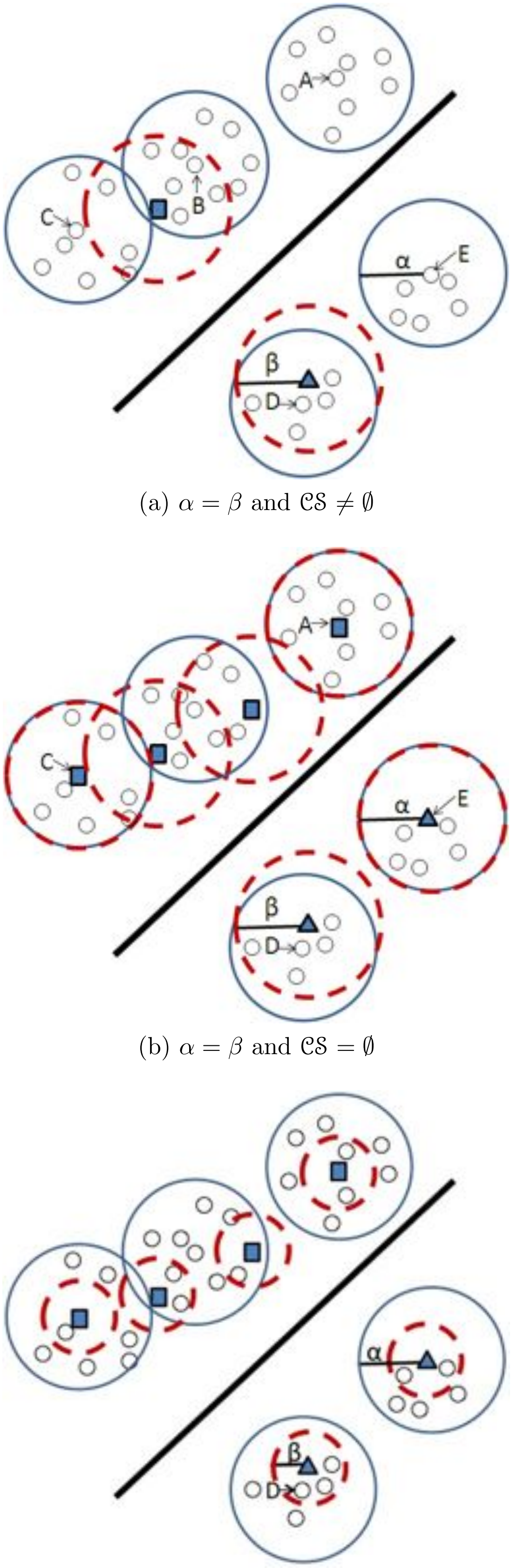
The work by Cebron & Berthold (2008) proposed a method considering the density and diversity information in selection, however they set the parameters in calculating density and diversity as positive constants which is a static way. Different from the static way of setting parameters as in Shen *et al.* (2004) and Cebron & Berthold (2008), we set the parameter β in a dynamical way. Initially, we set $\beta = \alpha$

as shown in Figure 6.2a, where shaded polygons represent labelled examples in $\mathcal{L}$ and circles represent unlabelled examples in $\mathcal{U}$. The regions defined by α are shown as solid circles for a small number of unlabelled examples (A, B, C, D and E). For clarity of illustration, rather than showing the regions defined by β around every unlabelled example, we show them, as broken circles, around only the labelled examples. The effect, however, is the same: if a labelled example is within the neighbourhood of an unlabelled example defined by β, then the unlabelled example will also be within the neighbourhood of the labelled example defined by β.

In the example shown in Figure 6.2a, since examples B and D have labelled examples in the neighbourhood defined by β, they will not be added to $\mathcal{CS}$. A, C and E, however, will be added. As more examples are labelled, we may reach a stage when there are no examples in the candidate set as there are always labelled examples within the neighbourhood defined by β. This scenario is shown in Figure 6.2b. When this happens we need to increase β to shrink this neighbourhood as shown in Figure 6.2c. We update β when we have no examples left in $\mathcal{CS}$ – a unique feature of our approach as far as we are aware.

We use a novel method to update β motivated by a desire to be able to set the size of $\mathcal{CS}$. As the size of the $\mathcal{CS}$ is defined by β, a bigger β value which defines a smaller neighbourhood gives us a bigger candidate set. We set β to a value which can give us a candidate set with a size proportional to the number of elements available for labelling (i.e. the size of the unlabelled pool $\mathcal{U}$) as detailed below:

(i) Calculate the similarity between each unlabelled example and its nearest labelled neighbour giving the set S, as follows



(a) $\alpha = \beta$ and $\mathcal{CS} \neq \emptyset$

(b) $\alpha = \beta$ and $\mathcal{CS} = \emptyset$

(c) $\alpha \neq \beta$ and $\mathcal{CS} \neq \emptyset$

Figure 6.2: The relationship between parameters α and β and the candidate set $\mathcal{CS}$.

$$S = \{s_i = \frac{1}{diversity(x_i)} \mid x_i \in \mathcal{U}\}$$

(ii) Sort the similarities s_i ($i = 0, 1, \ldots, n$) in S in ascending order and choose the value s_w from S that splits S into two, where

$$\begin{aligned} S_1 &= \{s_i \in S \mid s_i \leq s_w\}, \\ S_2 &= \{s_j \in S \mid s_j > s_w\} \text{ and} \\ |S_1| &= \lfloor (w \times |S|) \rfloor + 1, 0 \leq w \leq 1 \end{aligned}$$

(iii) Let $\beta = s_w$, which is the similarity value such that w proportion of unlabelled examples will be in diverse neighbourhoods of the feature space.

The proportion parameter, w, allows us to balance the influence of diversity and density in our selection strategy, namely the *balancing parameter*. When $w = 0$, the EGAL algorithm defaults to pure diversity-based sampling discounting any density information. As w increases, the influence of density increases and the influence of diversity decreases with more examples being added to $\mathcal{CS}$. When $w = 1$ the EGAL algorithm becomes purely a density-based sampling algorithm. We explore the effect of changing the value of the balancing parameter w in Section 6.3.1.

The procedure of EGAL is summarised in Algorithm 4 where the batch size b is set to one. EGAL can be implemented very efficiently. At the start the pairwise similarity matrix for the entire dataset and the individual density measure for every example are calculated and cached. At each iteration of the selection algorithm, the updated diversity measure for each example in the unlabelled set, $\mathcal{U}$, is the only calculation necessary. Computationally this is very efficient, especially considering the rebuilding of a classifier and the classification of every unlabelled example required by uncertainty sampling based methods at each iteration of the active learning selection.

```
Input: An initial labelled set $\mathcal{L}$, an unlabelled pool $\mathcal{U}$ of $n$ examples, a
       stopping criterion $\mathcal{SC}$, a batch size $b$, a balancing parameter $w$
Output: A labelled dataset
Compute the similarity matrix $\mathcal{M}$ of $s(i,k)$ where $x_i, x_k \in \mathcal{L} \cup \mathcal{U}$;
Set $\alpha = \beta = \mu - 0.5 \times \delta$; $\mu$ and $\delta$ being the mean and standard deviation of
the similarity matrix $\mathcal{M}$;
Calculate density for all the unlabelled examples $x_i, i \in I_u$ using Equation 6.1;
while $\mathcal{SC}$ is not met do
    $\mathcal{CS} = \emptyset$, $Selected = \emptyset$;
    Construct the candidate set $\mathcal{CS}$ as in Equation 6.4;
    while $|\mathcal{CS}| = 0$ do
        Update $\beta$ ;
        Update $\mathcal{CS}$ ;
    end
    Rank examples in $\mathcal{CS}$ by descending density order;
    foreach $t, t = 1 \ldots b$ do
        if $|\mathcal{CS}| < b$ then
            $Selected = Selected \cup \mathcal{CS}$ ;
        else
            Select the top $b$ ranked examples from $\mathcal{CS}$ with highest density and
            add them into $Selected$;
        end
    end
    Label each example $x_i \in Selected$ ;
    $\mathcal{L} = \mathcal{L} \cup Selected$ , $\mathcal{U} = \mathcal{U}/Selected$ ;
end
```

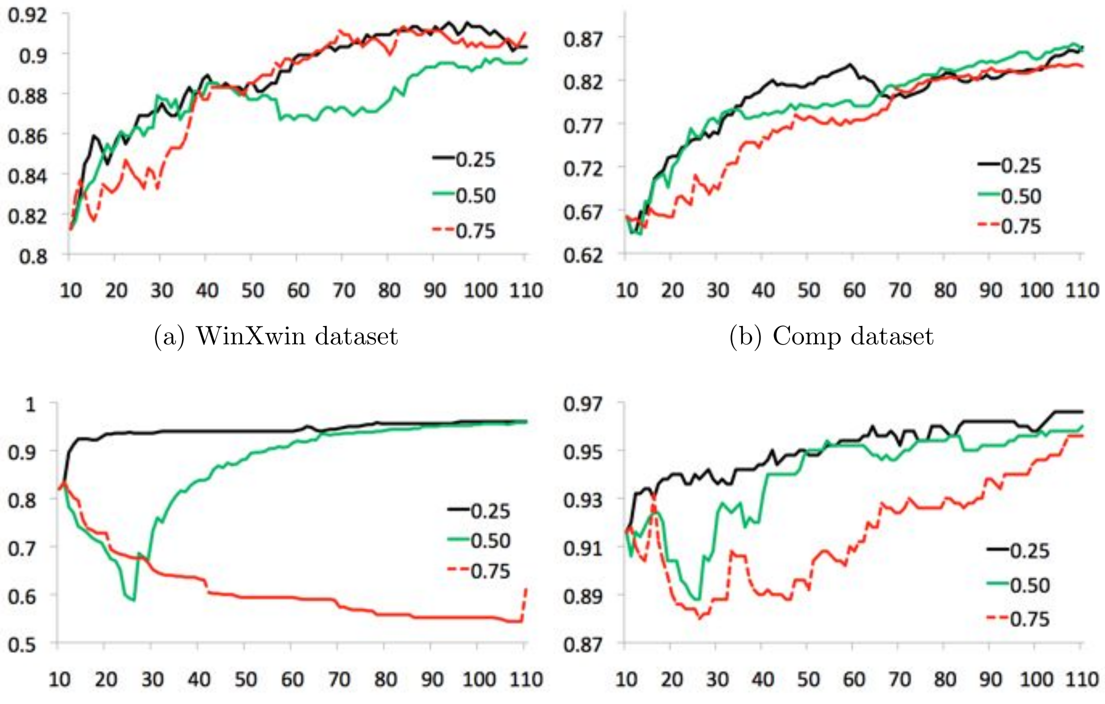
Algorithm 4: The EGAL algorithm

6.3 EGAL Evaluation

To assess the performance of our EGAL algorithm, we performed a comparative evaluation with other active learning selection strategies. The objective of our evaluation was firstly to see whether the performance of combining density and diversity information in our EGAL approach was better than density or diversity sampling alone. In addition, we compared EGAL to uncertainty sampling which is the most commonly used active learning selection strategy, and density-weighted uncertainty sampling which is the most common approach to combining density and uncertainty. The datasets used and the evaluation measures used are described in Section 4.2.1 and 4.2.3 respectively. The initial training set contains 10 examples selected for labelling by the oracle using a deterministic clustering approach – affinity propagation clustering (Chapter 7 and Hu *et al.*, 2010c). The same initial training set is used by each active learning algorithm for each dataset.

6.3.1 Role of the Balancing Parameter w

The density neighbourhood threshold, α, is set to $\mu - 0.5 \times \delta$ (as discussed in Section 6.2), as preliminary experiments showed it to be a good choice. In order to set the diversity neighbourhood threshold β, a value of w which controls the balance between density and diversity in the EGAL selection process is required. Intuition would suggest that diversity is more important than density, and in order to investigate this experiments were performed with w set to $0.25, 0.50$ and 0.75 on the datasets described previously. Results on four of these datasets are shown in



(a) WinXwin dataset

(b) Comp dataset

(c) Reuters dataset

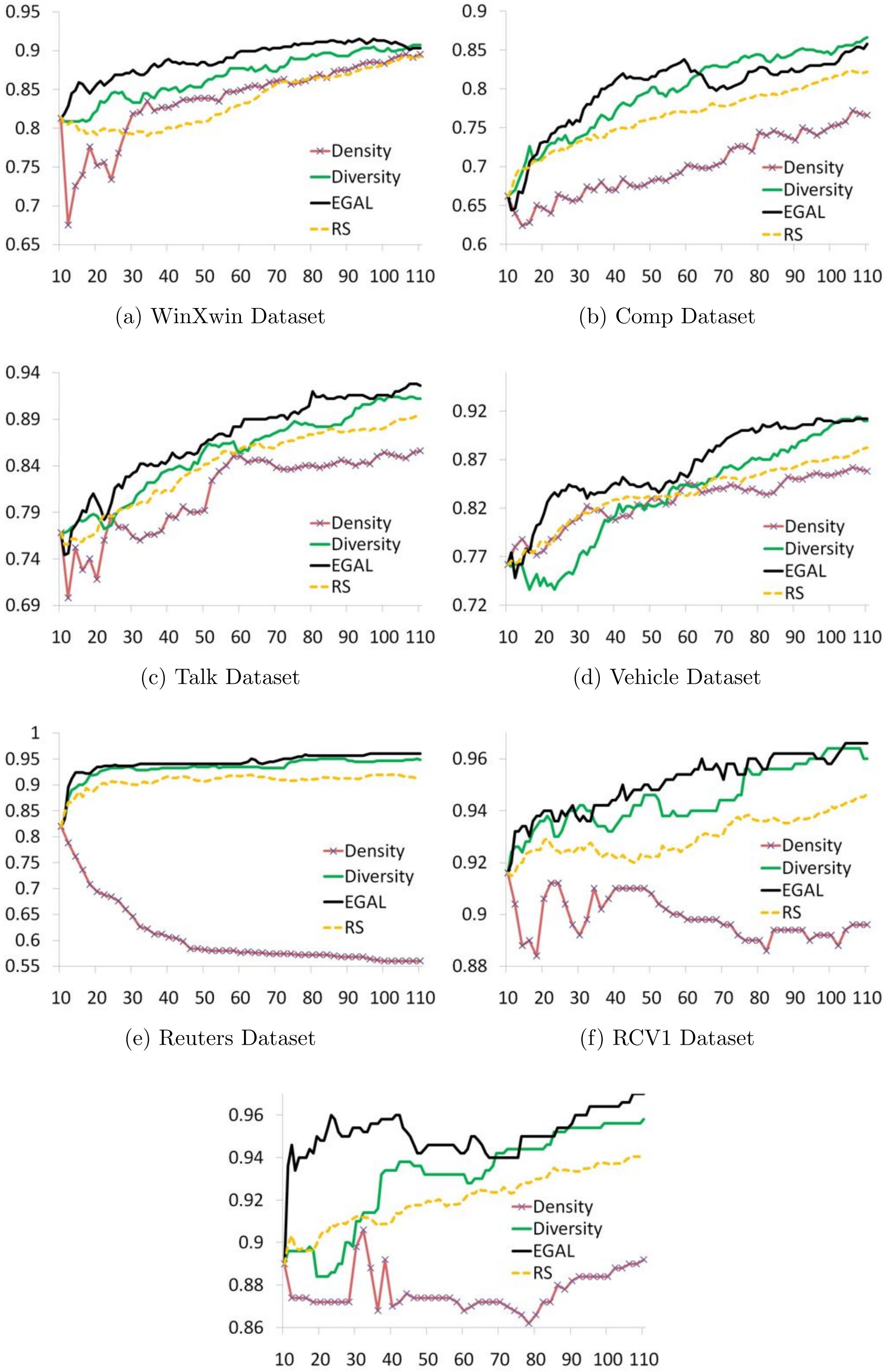
(d) RCV1 dataset

Figure 6.3: The effect of the balancing parameter w on the EGAL algorithm.

Figure 6.3. It was clear that $w = 0.25$ gave the best results (indicated by the fact that the learning curve for $w = 0.25$ dominates the others) and this value was used in all further experiments. This experiment supports the intuition that diversity is more important than density in the selection process.

6.3.2 EGAL Evaluation Results

The results of comparisons between our proposed approach (labelled EGAL), density sampling (labelled Density) and diversity sampling (labelled Diversity) across the seven datasets are summarised in Figure 6.4. A random sampling strategy (labelled RS) is used as a baseline. The results show that density sampling doesn't perform well but that diversity sampling performs consistently better than the baseline



(a) WinXwin Dataset

(b) Comp Dataset

(c) Talk Dataset

(d) Vehicle Dataset

(e) Reuters Dataset

(f) RCV1 Dataset

(g) Spam Dataset

Figure 6.4: Comparison of Density, Diversity, EGAL and RS selection strategies.

random sampling on six out of the seven datasets (except the Vehicle dataset). In addition, incorporating density information with diversity sampling in our EGAL algorithm improves the performance of diversity sampling consistently on all datasets. Table 6.1 shows the AULC scores of the four compared methods. The last row of Table 6.1 indicates the average rank of each method across the seven datasets and confirms the superiority of EGAL algorithm.

Table 6.1: AULC scores and ranks of density sampling, diversity sampling, random sampling and the EGAL algorithm.

	Density	Diversity	EGAL	RS
WinXwin	83.7(3)	86.7(2)	89.0(1)	83.5(4)
Comp	70.0(4)	79.7(2)	79.9(1)	76.6(3)
Talk	81.1(4)	85.4(2)	87.0(1)	84.1(3)
Vehicle	82.7(4)	83.4(3)	86.4(1)	83.5(2)
Reuters	60.6(4)	93.3(2)	94.3(1)	90.8(3)
RCV1	89.8(4)	94.5(2)	95.1(1)	93.0(3)
Sapm	87.7(4)	93.1(2)	95.1(1)	92.1(3)
Average Rank	**3.9**	**2.1**	**1.0**	**3.0**

We also compared EGAL to the more frequently used uncertainty sampling (US), as described in Chapter 4 on page 178. A density-weighted uncertainty sampling method (DWUS, Section 3.3.3.1) was also included as a comparison. In DWUS, the ranking score $r(x)$ of an example x is defined as in Equation 6.5 where uncertainty is multiplied with the density measure and examples with the highest resulting ranking score are selected for labelling. Furthermore, a confidence-based selection strategy – ACMS is included in the comparison. The graphs of learning curves are shown in Figure 6.5.

$$r(x) = density(x) \times uncertainty(x) \tag{6.5}$$

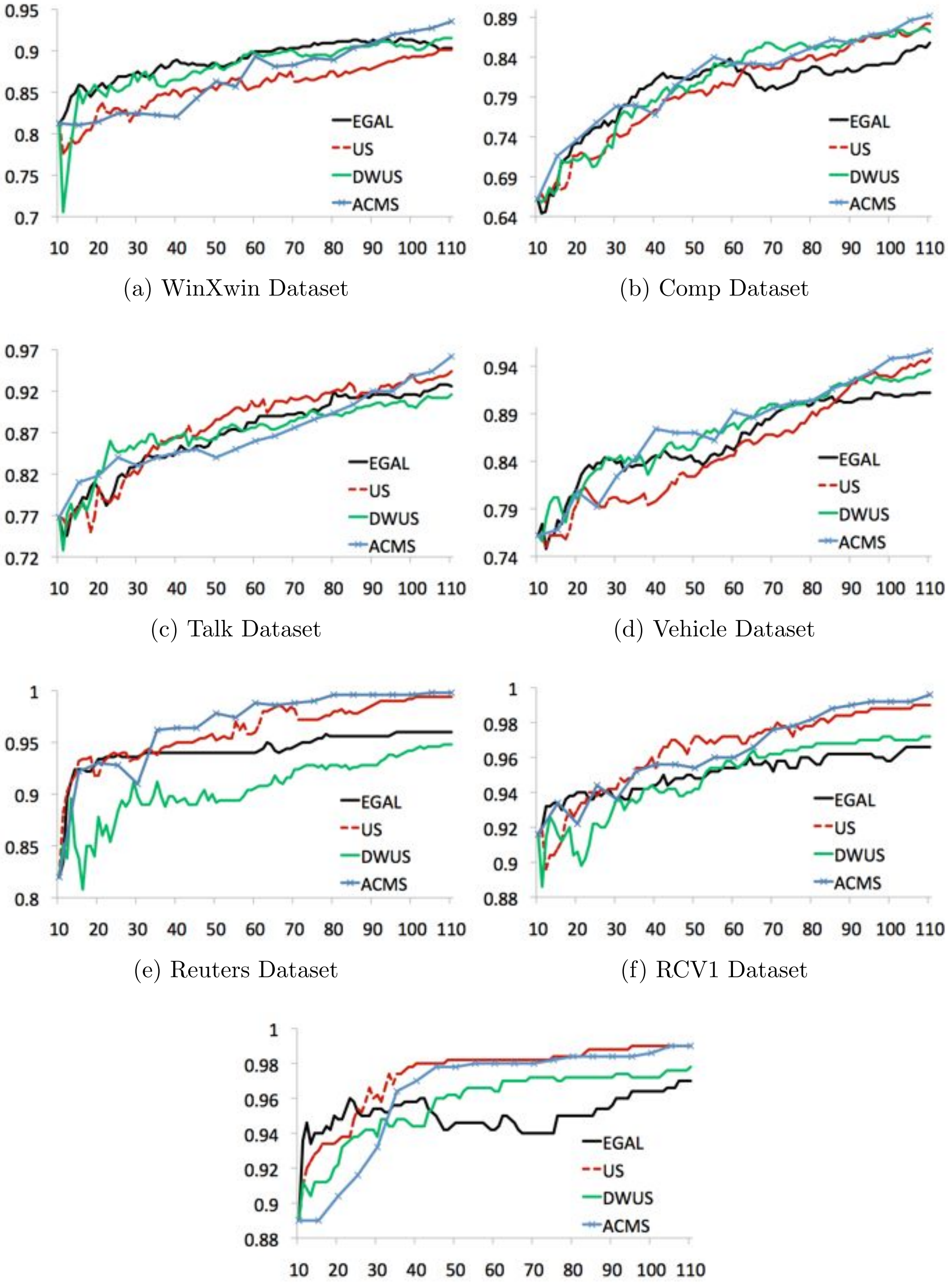
Previous work on density weighted uncertainty sampling has shown an improvement over uncertainty sampling (Nguyen & Smeulders, 2004; Settles & Craven, 2008). As expected results in Figure 6.5 is in agreement with results observed on datasets suited to density algorithms. However, for datasets where density sampling performs badly (see Figures 6.5e, 6.5f and 6.5g) DWUS does not improve performance over US indicating that the density information is having a negative effect on the active learning process.

The more interesting benefit of EGAL is in the early stage of the active learning process, the first 20 to 30 labellings, where it outperforms US, DWUS and ACMS. A detailed analysis of the AULC scores for learning curves up to a varying number of labels was performed. Illustrative examples of AULC values are given in Table 6.2. The difference between US and EGAL was found to be significant (at $\alpha = 0.05$) using the Wilcoxon signed-rank test at 30 labels and below. Even when we look at more sophisticated confidence based selection strategy – ACMS, EGAL is still good at the beginning of the labelling stage with the best average rank of 1.7 as shown in bold font in the last row of Table 6.2. There was no significant difference between DWUS and US at any other number of labels. It can be seen that EGAL performs poorly on three datasets (Talk, Vehicle and Retuers) even at the early learning stage ($\mathcal{SC} \leq 30$). One possible reason is that density sampling performs worse than random sampling on all of these three datasets especially on the Reuters dataset.

These results point towards an interesting empirical property of the EGAL algorithm: it can improve the labelling accuracy fastest in the beginning stages of active learning.

Table 6.2: Illustrative AULC values for learning curves up to the specified number of labels. The best values across the four approaches are highlighted in bold.

Dataset	$\mathcal{SC} = 30$				$\mathcal{SC} = 60$				$\mathcal{SC} = 110$			
	US	DWUS	ACMS	EGAL	US	DWUS	ACMS	EGAL	US	DWUS	ACMS	EGAL
WinXwin	16.2	16.7	16.3	**17.1**	41.8	43.0	41.6	**43.6**	85.7	88.1	86.8	**89.0**
Comp	14.0	14.0	**14.6**	14.4	37.5	38.0	38.6	**38.7**	79.9	80.9	**81.4**	79.9
Talk	15.7	16.2	**16.3**	15.9	41.9	**42.2**	41.7	41.6	**87.9**	87.0	87.0	87.0
Vehicle	15.7	**16.2**	15.8	16.1	40.2	**41.8**	41.7	41.4	85.3	87.4	**87.6**	86.4
Reuters	**18.5**	17.3	18.3	18.4	47.0	44.2	**47.3**	46.6	96.2	90.6	**97.0**	94.3
RCV1	18.5	18.3	18.6	**18.7**	**47.4**	46.6	47.2	47.1	**96.5**	94.9	96.4	95.1
Spam	18.8	18.5	18.1	**18.9**	**48.1**	47.2	47.3	47.4	**97.4**	95.8	96.5	95.1
Average Rank	3.0	2.7	2.4	**1.7**	**2.4**	2.7	**2.4**	**2.4**	2.3	2.7	**1.7**	2.7



(a) WinXwin Dataset

(b) Comp Dataset

(c) Talk Dataset

(d) Vehicle Dataset

(e) Reuters Dataset

(f) RCV1 Dataset

(g) Spam Dataset

Figure 6.5: Comparison of EGAL, US, DWUS and ACMS selection strategies.

6.4 Examples of Visualised Active Learning

This section demonstrates how a visualisation technique – CBTV-AL (Mac Namee *et al.*, 2010) can be used to visualise the active learning process on the seven datasets, and describes how this offers insight into the understanding of different selection strategies.

Four selection strategies including density sampling, diversity sampling, EGAL and uncertainty sampling are visualised. Snapshots of the four selection strategies on the WinXwin dataset and the Reuters dataset are shown from Figure 6.6 to Figure 6.13 which give a visual insight of the active learning process. A summary of the snapshots is shown in Table 6.3.

In all the figures, examples labelled by the oracle are shown enlarged, and both the colours and shapes of these labelled examples represent their *true class*. In the graphs for density sampling, the darkness of each unlabelled example indicates its density (darkness increases with density). Similarly, the darkness in the graphs for diversity sampling shows the diversity scores for the examples in the pool and the darkness in the graphs for uncertainty sampling shows the uncertainty scores of unlabelaled examples. In the graphs for EGAL, the darkness represents the diversity as in the graphs for diversity sampling. The first subgraph in each figure, such as Figure 6.6a, shows the initial stage of the active learning process where 10 initial examples are selected using a deterministic clustering approach – agglomerative hierarchical clustering. The following subgraphs in each figure show the dataset after every 35 examples getting labelled by using a specific selection strategy. The last subgraph in each figure shows the dataset with all the examples in it getting labelled which shows the real class distribution of the dataset.

Table 6.3: Snapshots of Density Sampling, Diversity Sampling, EGAL and Uncertainty Sampling on the Seven Datasets.

	Density Sampling	Diversity Sampling	EGAL	Uncertainty Sampling
WinXwin	Figure 6.6	Figure 6.7	Figure 6.8	Figure 6.9
Reuters	Figure 6.10	Figure 6.11	Figure 6.12	Figure 6.13

Figure 6.6 shows snapshots of the active learning labelling process running on the WinXwin dataset using density sampling from 10 initially labelled examples to the whole dataset getting labelled. According to the description of the density sampling approach (Section 6.2.1), the density measures remain constant throughout the active learning process which is also confirmed in the graphs for density sampling. It also can be seen that examples are denser in the centre region. The same effect holds for the Comp, Talk, Vehicle and RCV1 datasets. It is interesting to note the selection path which indicates that examples in the centre region are labelled first moving then to the outside examples. This is different for the Reuters dataset (Figure 6.10) where examples from one class is more densely packed than the other class. Accordingly, examples located in much denser area (green circle examples in Figure 6.10) are continuously selected to label while examples from the other class are ignored until almost all denser green circle examples getting labelled. This illustrates why density sampling is a myopic selection strategy and can result in very poor performance as discussed in Section 6.2.3.

Figure 6.7 shows a series of snapshots of the results of the active learning process running on the WinXwin dataset using diversity sampling. It can be seen that the selection path of diversity sampling is from outside area to the inside area which is the opposite to density sampling. The similar effect holds on the Comp, Talk, Vehicle and RCV1 datasets.

Figure 6.8 shows snapshots of the same process using EGAL. As described in Section 6.2.3, EGAL selects the most informative examples with high density and diversity at the same time. The selection path of EGAL is more balanced. It can be seen that the EGAL approach achieves a much more successful balance between exploration and exploitation and selects examples for labelling around the full example space. The selection path of EGAL is very similar to that of diversity sampling. As described in Section 6.3.1, the balancing parameter w is 0.25 which gives much higher weight to diversity than density. The same effect holds for EGAL on the other six datasets that EGAL selects more balanced examples from a wider example space and the selection path of EGAL is more similar to that of diversity sampling.

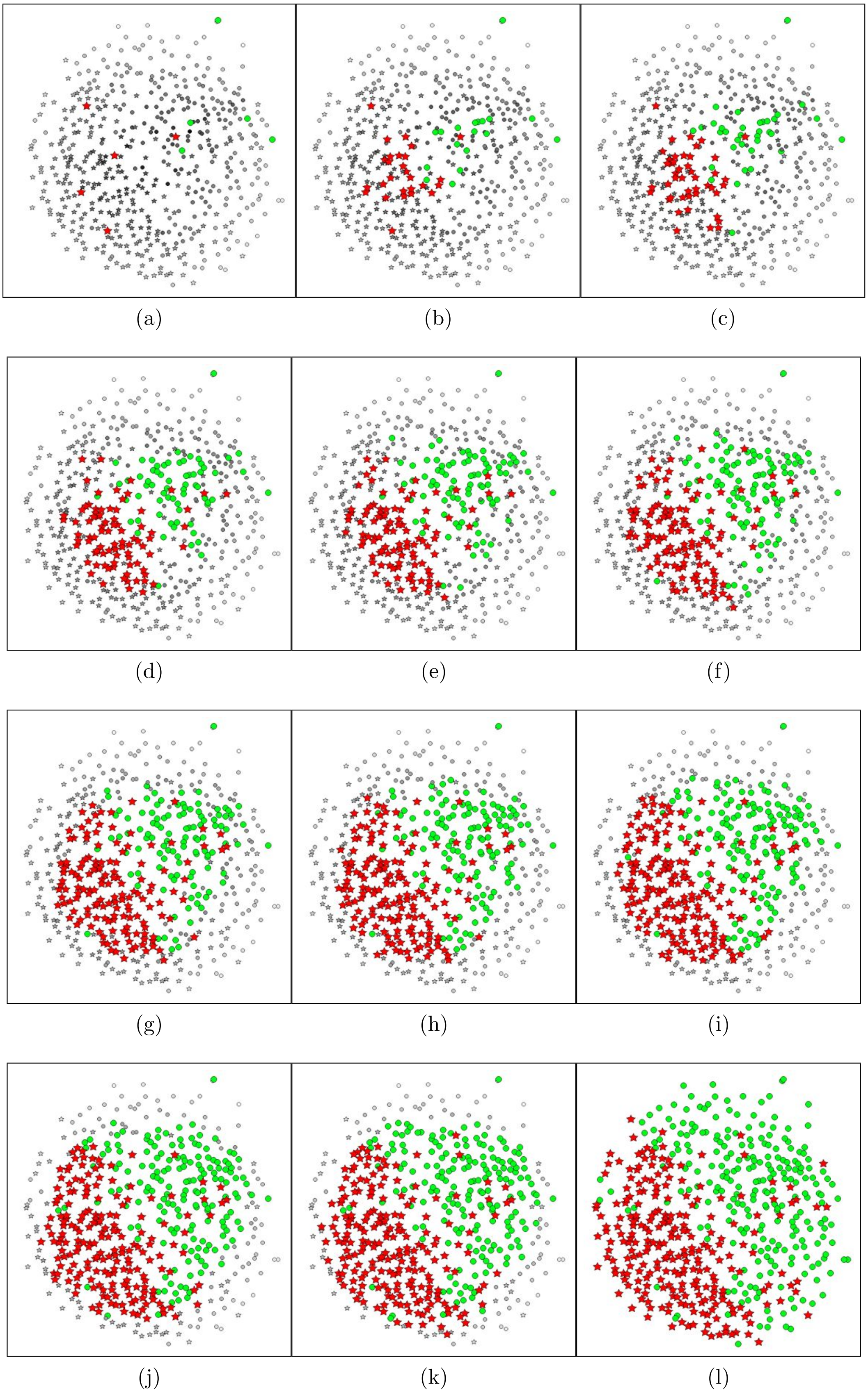
Figure 6.9 shows the snapshots of using uncertainty sampling on the WinXwin dataset. Uncertainty sampling concentrates on the most uncertain examples which are also examples closest to the classification boundary which is confirmed in Figure 6.9. This conclusion is more clearly supported by the Reuters dataset which is a linearly separable dataset and the visualisation indicates that classification uncertainty is centred on the border between the two classes. For the non-linearly separable datasets (WinXwin and other four datasets), it is harder to find the classification boundary so the selection path is not as clear as that in the linearly separable datasets.

From the above figures, it can be seen that visualisation techniques can be used to offer insight into the inner workings of active learning selection strategies which makes it possible to help developers create more effective active learning systems.

6.5 Conclusions

In this chapter, we have proposed EGAL, an exploration-only approach to active learning-based labelling of datasets. EGAL uses only the notions of density and diversity, based on similarity, in its selection strategy. This avoids the drawbacks associated with exploitation-based approaches to selection. Furthermore, in contrast to most active learning methods, because EGAL does not use a classifier in its selection strategy it is computationally efficient. We have shown empirical results of EGAL's viability as a useful tool for building labelled datasets, especially in domains where it is desirable to front-load the active learning process so that it performs well in the earlier phases – a feature of EGAL demonstrated in our evaluation experiments.

Finally, we use visualisation techniques to support understanding and analysis of active learning and show that visualisations are powerful exploratory tools to effectively analyse the active learning process.



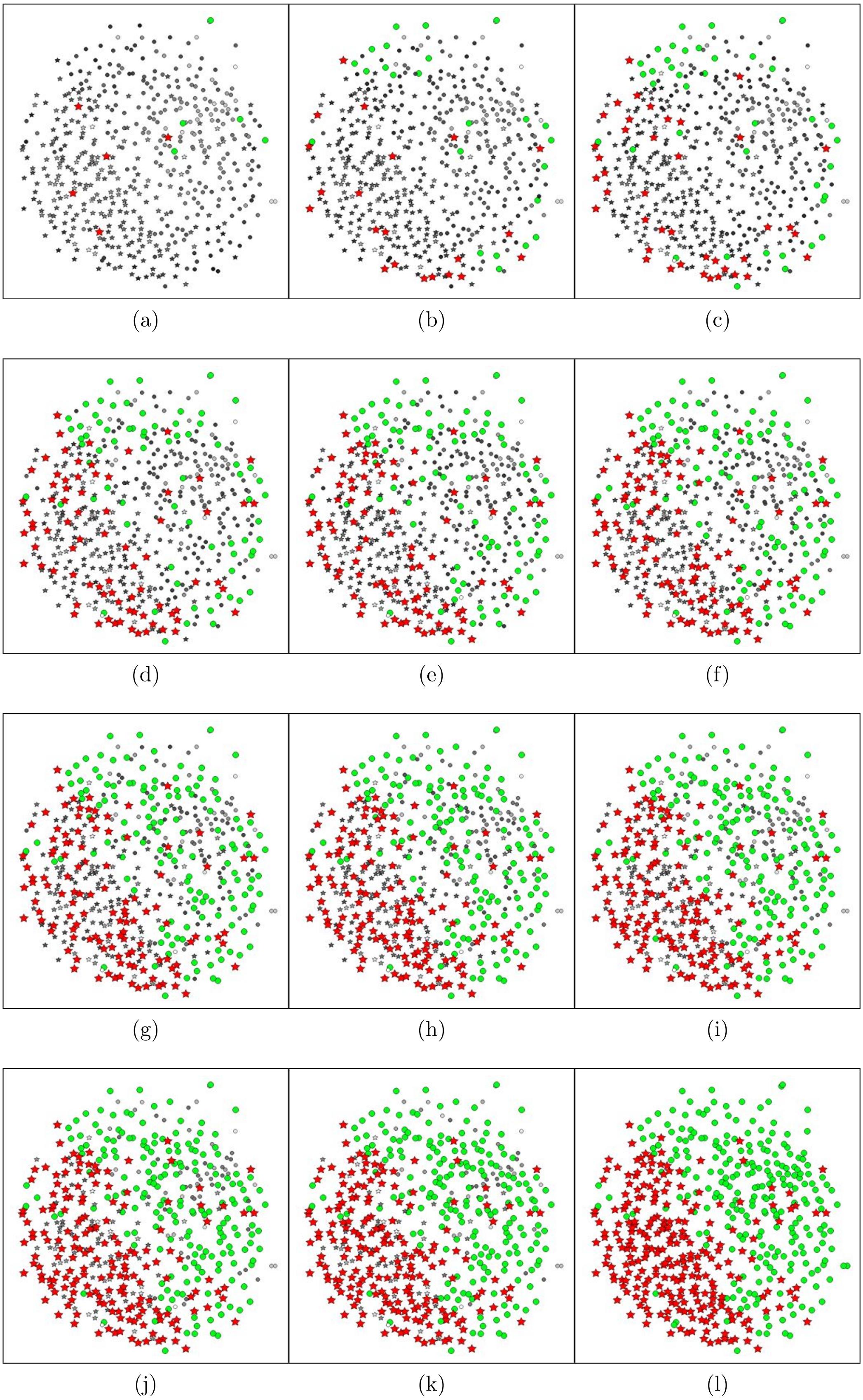
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 6.6: A visualisation of the active learning process running on the WinXwin dataset using a density only selection strategy.



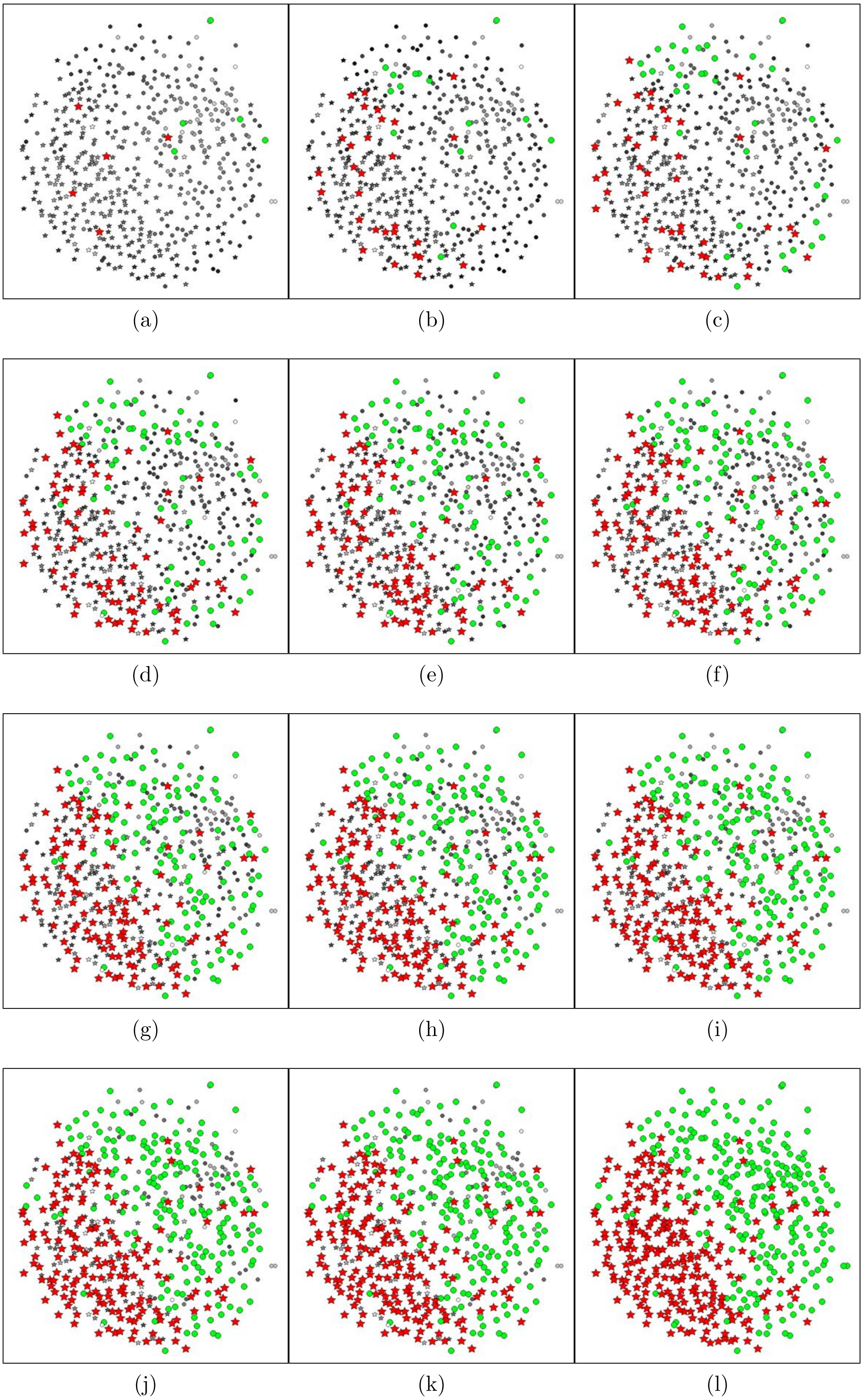
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 6.7: A visualisation of the active learning process running on the WinXwin dataset using a diversity only selection strategy.



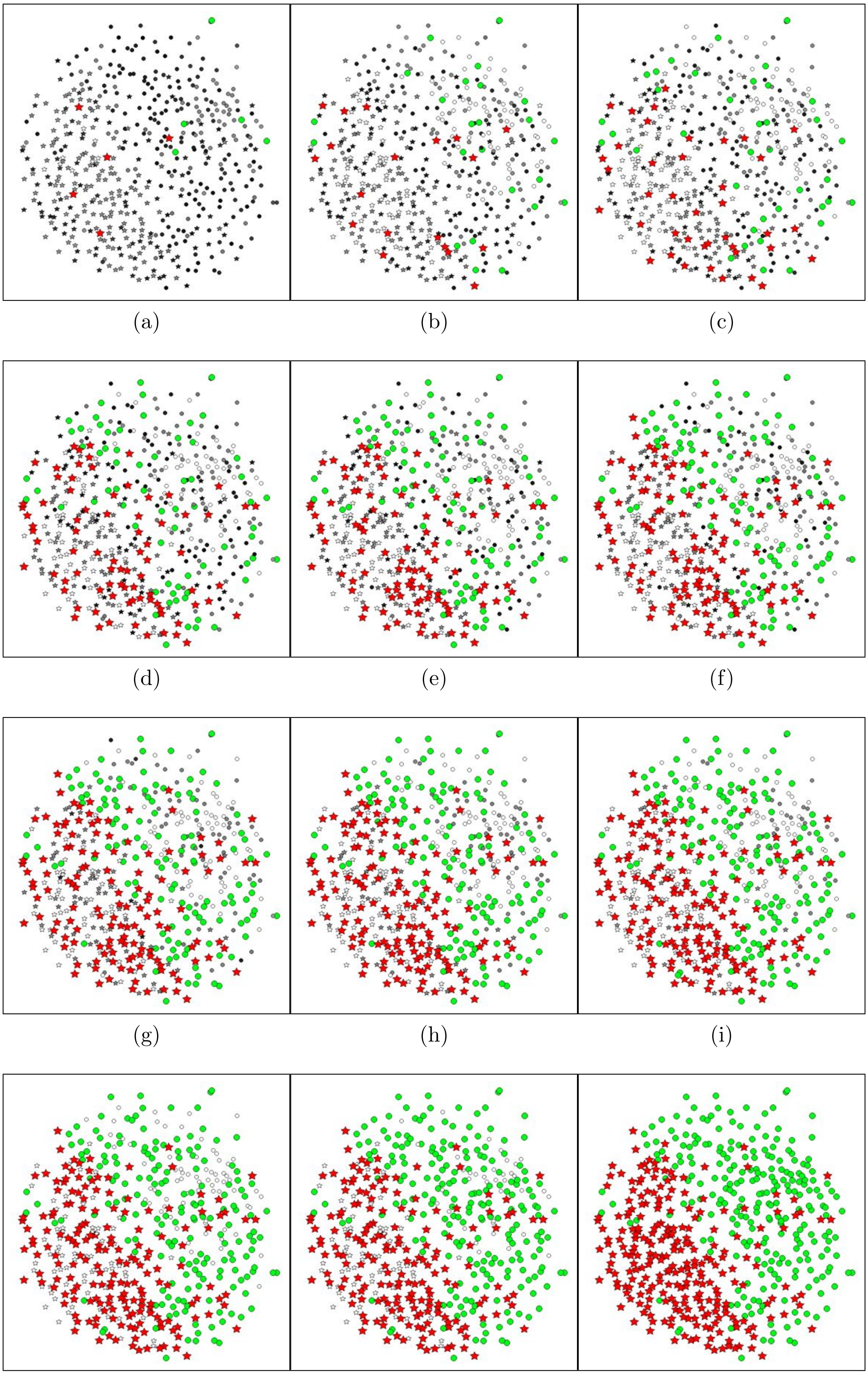
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 6.8: A visualisation of the active learning process running on the WinXwin dataset using the EGAL selection strategy.



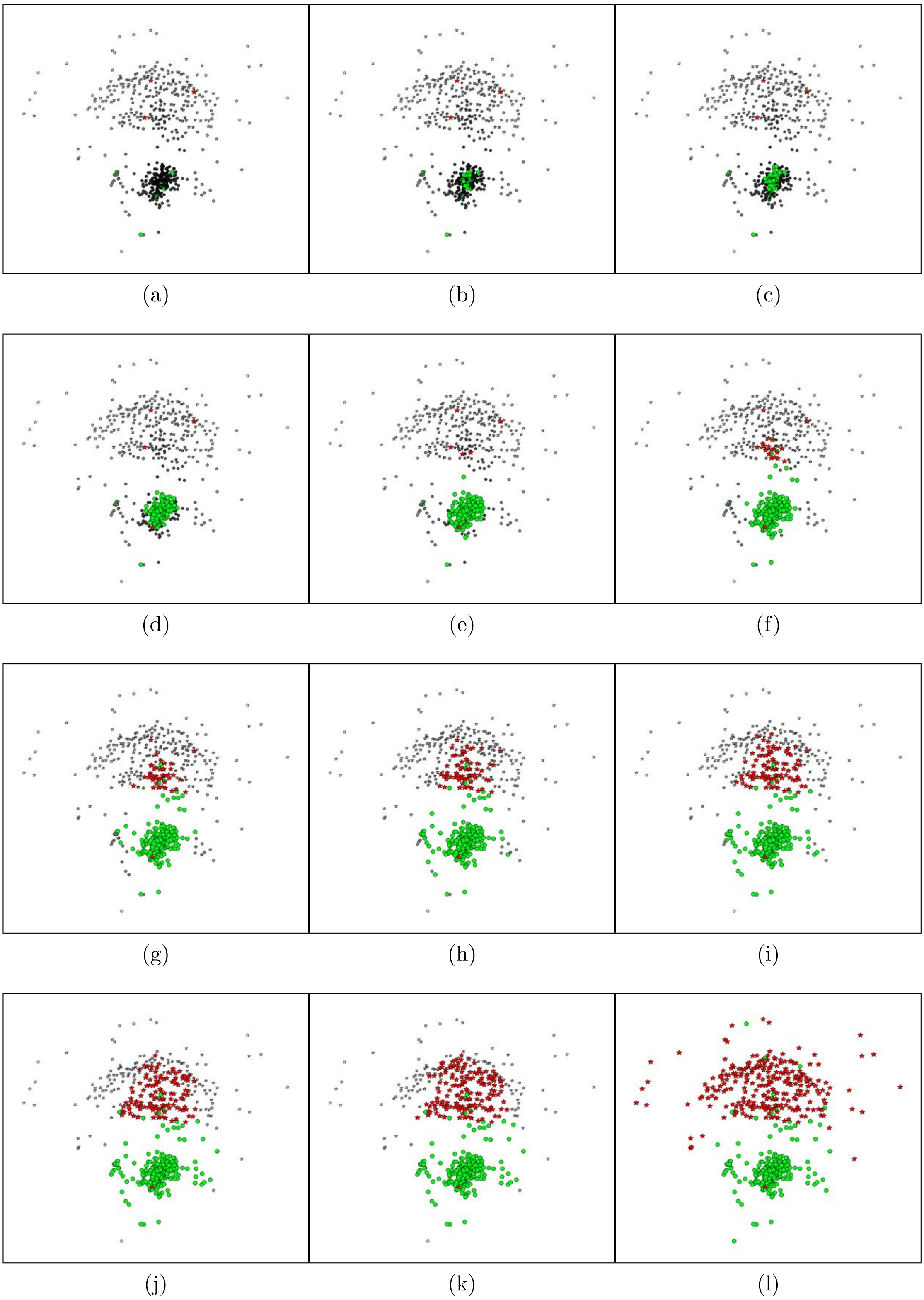
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 6.9: A visualisation of the active learning process running on the WinXwin dataset using an uncertainty sampling selection strategy.



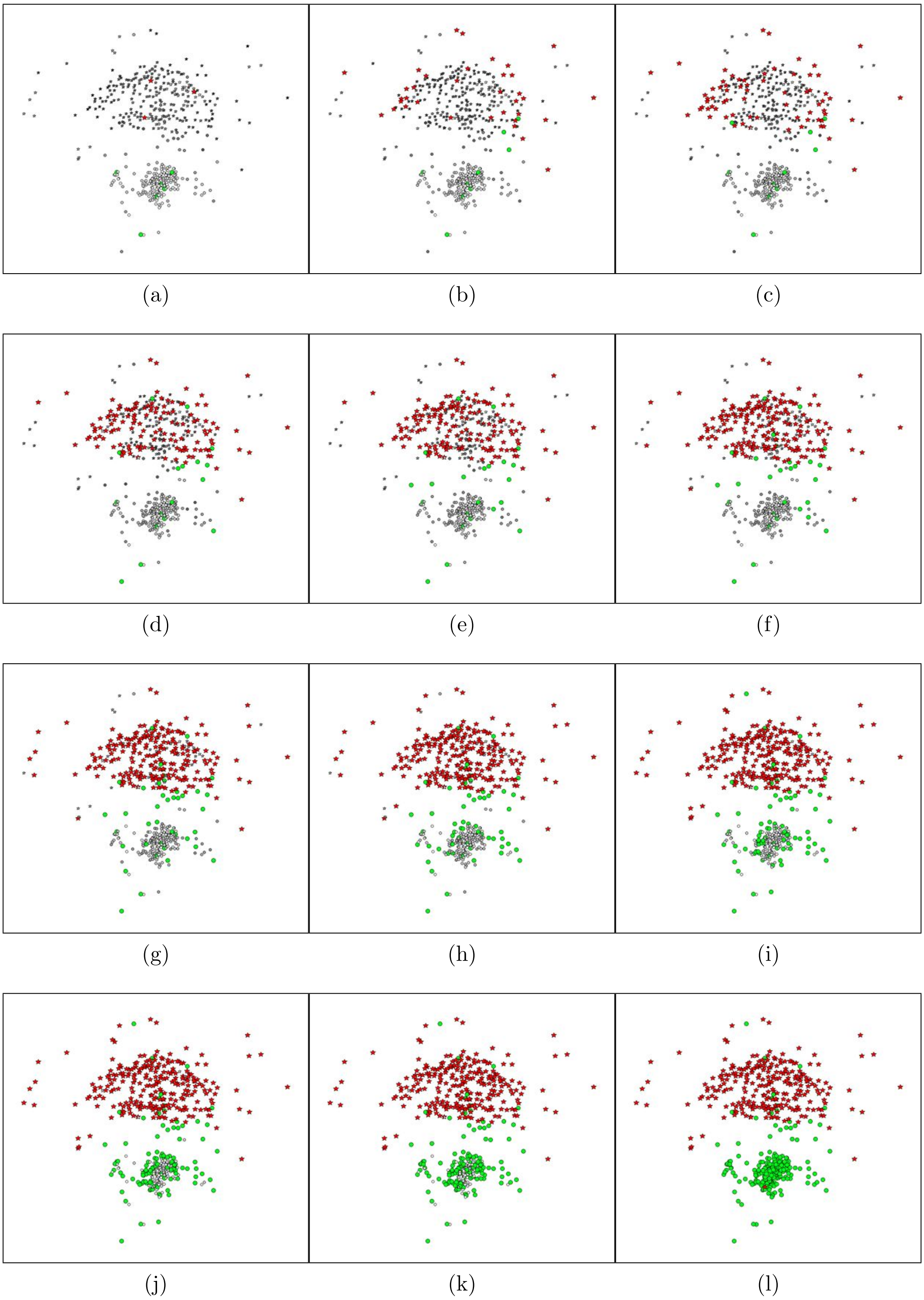
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 6.10: A visualisation of the active learning process running on the Reuters dataset using a density only selection strategy.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 6.11: A visualisation of the active learning process running on the Reuters dataset using a diversity only selection strategy.

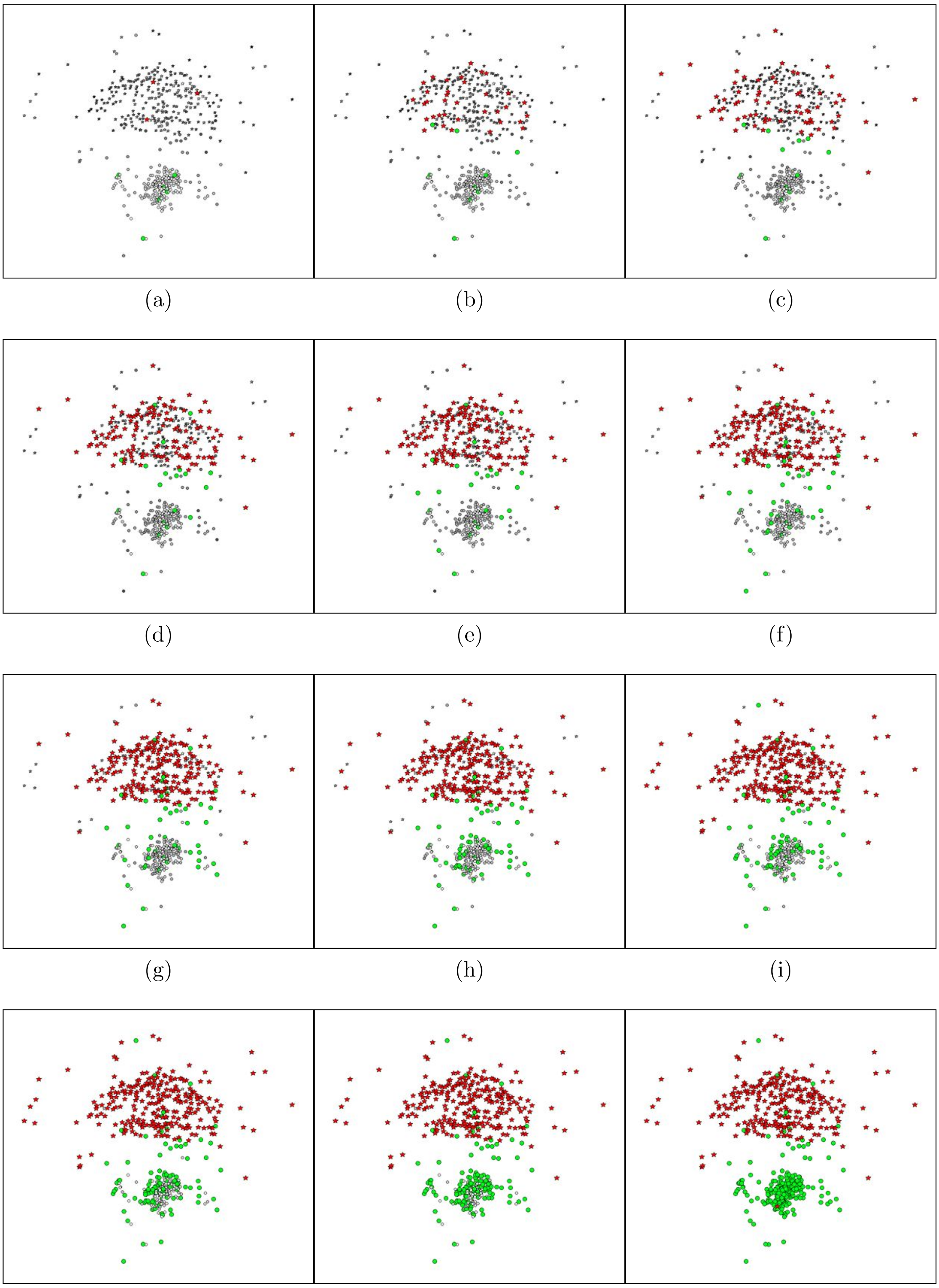
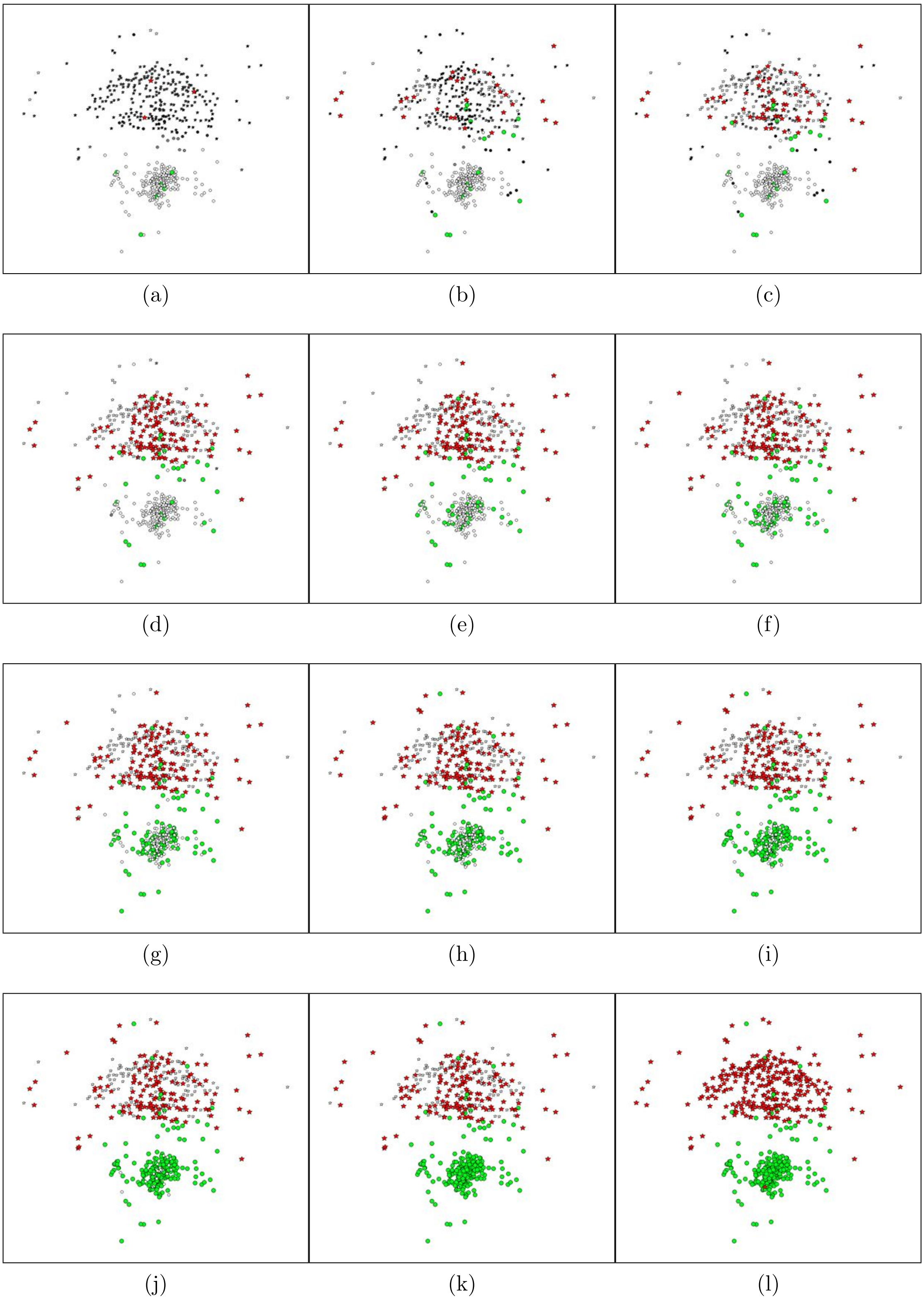


Figure 6.12: A visualisation of the active learning process running on the Reuters dataset using the EGAL selection strategy.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 6.13: A visualisation of the active learning process running on the Reuters dataset using an uncertainty sampling selection strategy.

Chapter 7

Initial Training Set Construction Using Deterministic Clustering

As described in Chapter 3, a small labelled set is needed to seed the active learning process. While the common approach for initial training set selection in active learning is to select the initial training examples randomly, better and more reliable performance can be achieved by taking the opportunity to seed the active learning process with carefully selected examples. This can be done using clustering techniques. In this chapter, we deal with initial training set construction for active learning. Existing work uses non-deterministic clustering to build the initial training set. We demonstrate a problem of using non-deterministic clustering methods to select initial training set in active learning and moreover we propose novel initial training set construction methods, three deterministic clustering algorithms that have not been considered in the context of initial training set selection before.

7.1 Introduction

The question of how best to populate the initial training set has received little consideration in the active learning community. In fact, most approaches ignore the problem and randomly choose examples.

In a review of 206 active learning papers from conferences including *NIPS*, *ICCV*, *CVPR*, *ICML*, *UAI* and *ECML*; journals including *Machine Learning*, *Pattern Recognition*, and *Data Mining & Knowledge Discovery*; and technical reports, over 94% of researchers use a randomly selected initial training set or failed to specify their initial training set selection method. Fewer than 6% used a targeted approach to populating their initial training set. This ignores an opportunity to improve on the effectiveness of the active learning process.

As reviewed in Section 3.2, to populate the initial training set in a more targeted way, clustering techniques can be used and in the active learning literature there is work that takes this approach, typically using k-Means, or k-Medoids. However, both the k-Means and k-Medoids algorithms are non-deterministic and "*can often lead to highly inconsistent results over many trials*" (Greene, 2006). This causes inconsistent performance when running the same active learning system on the same dataset several times, and so comparison results can be unreliable. This problem is exacerbated in text classification as the datasets are of extremely high-dimensionality which leads to considerable variability in the clustering results.

In this chapter we first illustrate the problems with using non-deterministic clustering for selecting initial training sets, showing how non-deterministic clustering methods can result in inconsistent behaviour in the active learning process, and we propose the use of deterministic clustering techniques to populate the initial training set. We then compare various deterministic clustering techniques and the non-deterministic ones, and show that deterministic clustering algorithms are as good as non-deterministic clustering algorithms at selecting initial training examples for the active learning process. More importantly, we show that the use of deterministic approaches stabilises the active learning process.

7.2 Evaluation

This section reports on the experiments performed to validate the use of deterministic clustering algorithms in selecting initial training sets for active learning. There are three objectives to the evaluations described here. The first is to confirm that better performance can be achieved by running active learning on an initial training set selected using clustering algorithms rather than one selected randomly. The second is to show that k-Means (Appendix C.1.1), k-Medoids (Appendix C.1.3) and KMeans+ME (Kang *et al.*, 2004) are non-deterministic and the impact of this on the active learning process. The third is to compare three deterministic clustering algorithms: *Furthest-First-Traversal* (FFT, Appendix C.1.4) , *Agglomerative Hierarchical Clustering* (AHC, Appendix C.1.5) and *Affinity Propagation Clustering* (APC, Appendix C.1.6); and confirm their superiority in selecting initial training examples.

Four textual datasets (WinXwin, Comp, Reuters and RCV1) as described in

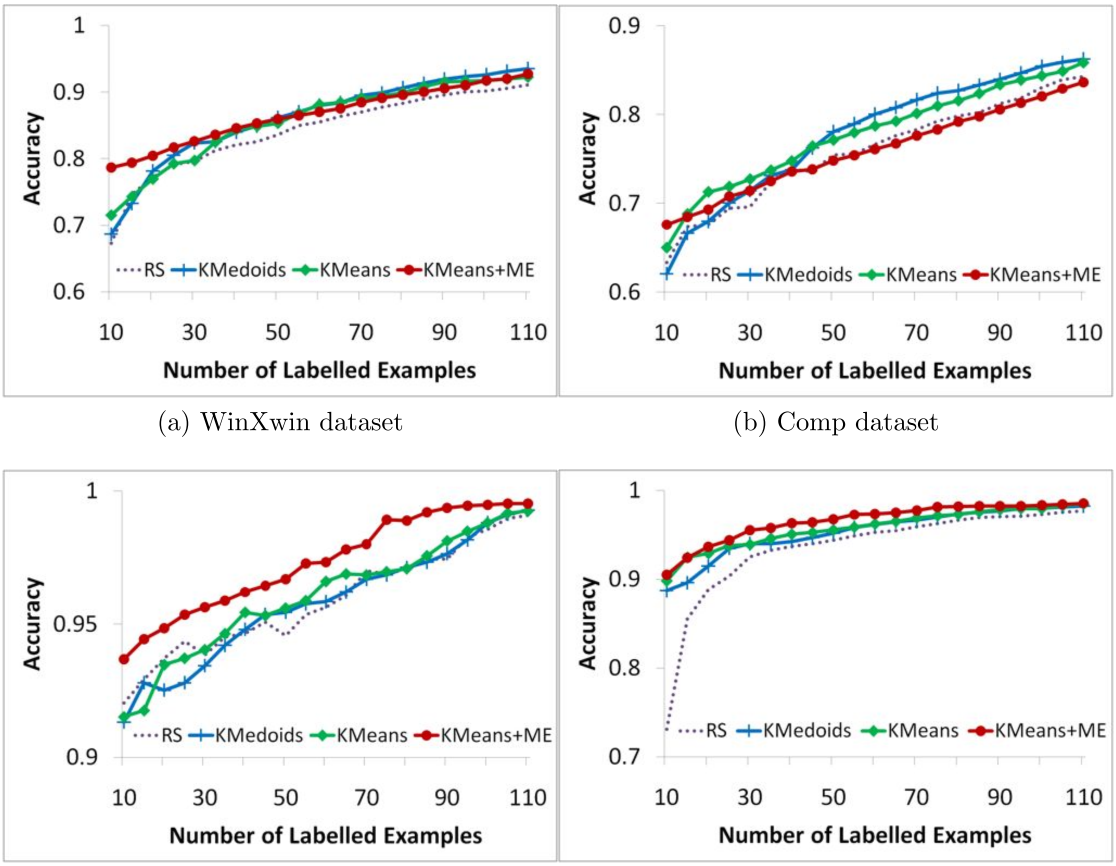
Section 4.2.1 (see Table 4.1 on page 83) are used in the evaluation. For each of the algorithms under consideration the active learning process was run to completion using the selected clustering technique to populate an initial training set of size 10. For those algorithms containing a random component (i.e. random selection, *k*-Means, *k*-Medoids and KMeans+ME) the process was repeated multiple times and average results are presented. The effectiveness of clustering-based initial training set is measured by comparing the ranks of the AULC scores and the learning curves of active learning using the clustering-based initial training set and the randomly picked initial training set. In our implementation of clustering algorithms, the cosine similarity based distance is adopted to measure the distance between examples and their centroids (medoids) or exemplars.

7.2.1 Comparison of Initial Training Sets Building Methods Using Clustering and Random Sampling

Initially the objective is to compare the performance of initial training sets selection methods using previously employed non-deterministic clustering techniques to see if a clustering approach can help to seed the active learning process. Three methods are involved: *k*-Means, *k*-Medoids and KMeans+ME. Methods using *k*-Means and *k*-Medoids clustering are picked because they are the most widely used initial training set selection methods with clustering. KMeans+ME is chosen in this study because it is the first well studied initial training set selection method and has shown better performance than the method using *k*-Means clustering. Initial training sets generated by these techniques are compared against a randomly selected initial training set.

Figure 7.1 shows the learning curves for random initial training set selection, and initial training set selection methods using k-Means, KMeans+ME, and k-Medoids clustering on the four datasets. From the learning curves in Figure 7.1, it is clear that the learning curves of methods using clustering tend to dominate that achieved when the initial training examples are selected randomly, and that amongst the clustering techniques the learning curve from the initial training set selected using KMeans+ME tends to dominate the others on two datasets as seen in Figure 7.1c and 7.1d. KMeans+ME doesn't performs well on the Comp dataset as seen in Figure 7.1b. One possible reason could be that cluster centers from k-Means on the Comp dataset do not work well as initial training examples (as seen in Figure 7.1b) and the introduction of the extra *model examples* makes it worse. One interesting thing to be noted from the learning curves is that the curve of KMeans+ME is flatter than the others due to the use of model examples indicating that the extra model examples smooth the learning process.

Based on the learning curves, AULC scores for each method are calculated on each dataset as shown in Table 7.1 with their ranks in parentheses. The last column of Table 7.1 shows the averaged ranks across the four datasets. From the results, it can be seen that when random selection is used to select the initial training set, it is ranked as 4 on three out of the four datasets and 3.5 across the four datasets which is worse than any of the clustering techniques used to select the initial training set. This confirms the superiority of using clustering in selecting the initial training examples. The second conclusion is that KMeans+ME is better than both k-Means



(a) WinXwin dataset

(b) Comp dataset

(c) Reuters dataset

(d) RCV1 dataset

Figure 7.1: The learning curves produced by the active learning process when the initial training set is chosen using random selection (RS), k-Means, KMeans+ME, and k-Medoids. Axes are zoomed for resolution.

Table 7.1: AULC scores of active learning using different non-deterministic clustering algorithms in initial training set selection.

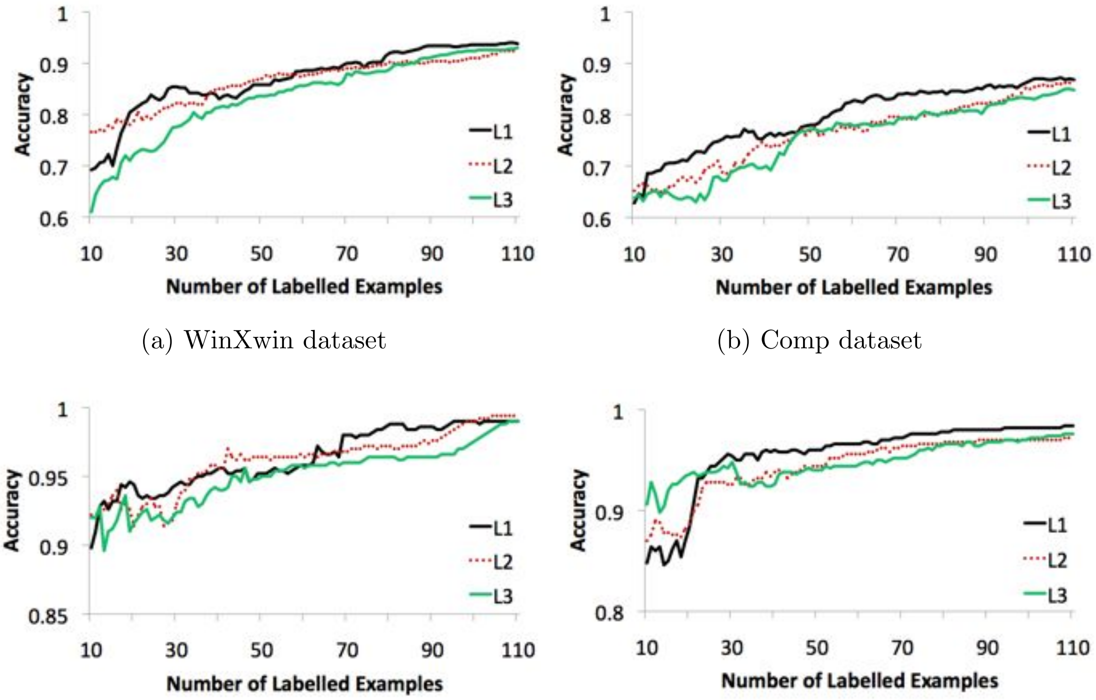
	WinXwin	Comp	Reuters	RCV1	Average Rank
RS	84.6(4)	76.1(3)	95.9(3)	94.1(4)	3.5
KMedoids	86.5(2)	78.1(1)	95.8(4)	95.4(3)	2.5
KMeans	86.0(3)	78.0(2)	96.1(2)	95.9(2)	2.25
KMeans+ME	86.6(1)	76.0(4)	97.3(1)	96.7(1)	1.75

and k-Medoids clustering with the highest averaged rank of 1.75 which confirms results presented in Kang *et al.* (2004) that KMeans+ME is better than using k-Means clustering only.

7.2.2 Illustrating the Impact of Non-Determinism

The second set of experiments sought to illustrate the impact of the non-determinism of the k-Means, KMeans+ME and k-Medoids clustering techniques. The active learning process was run repeatedly using each of these algorithms to select the initial training examples. Because the initial cluster centres are selected randomly each time, the initial training set for the same dataset is different on subsequent runs. This results in differing performance for the active learning process on each run.

Five runs are executed to generate five initial training datasets on each of the four datasets. Then the active learning process is seeded by each of the initial training set. Labelling accuracies are recorded and learning curves are plotted for each run. One run with high performance (labelled as 'L1'), one run with medium performance (labelled as 'L2') and one run with low performance (labelled as 'L3') are shown. Figure 7.2 shows the results of active learning using initial training sets



(a) WinXwin dataset

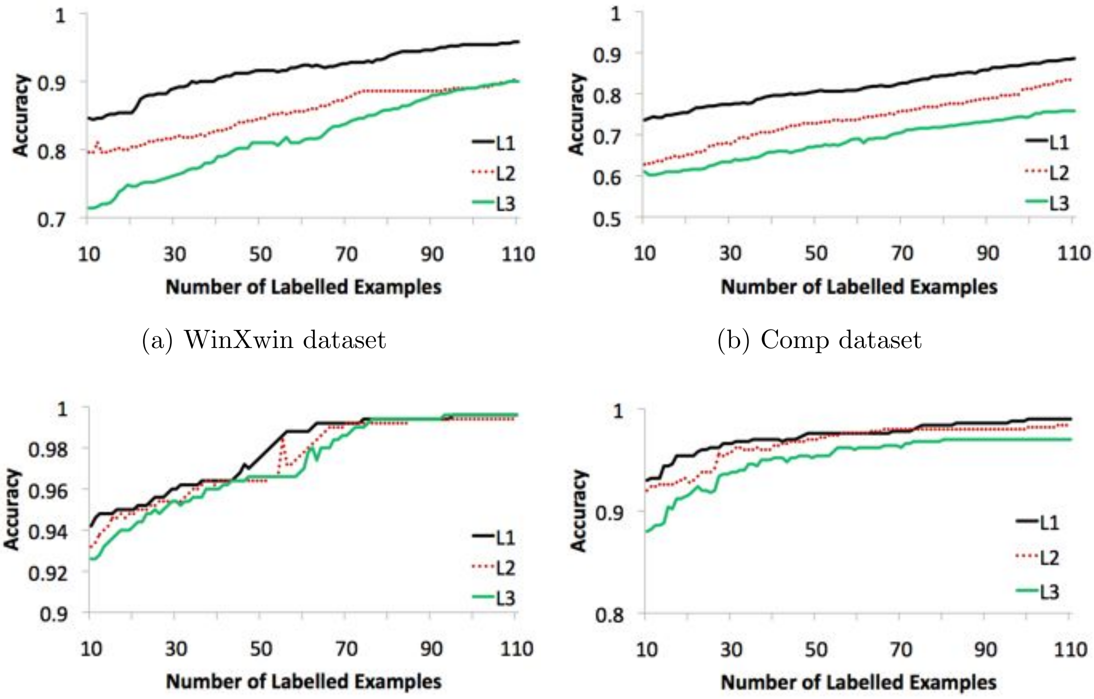
(b) Comp dataset

(c) Reuters dataset

(d) RCV1 dataset

Figure 7.2: The learning curves produced by three runs of the active learning process when the initial training set is populated using k-Medoids clustering. Axes are zoomed for resolution.

created by k-Medoids clustering on WinXwin (Figure 7.2a), Comp (Figure 7.2b), Reuters (Figure 7.2c) and RCV1 (Figure 7.2d) datasets. It is evident from Figure 7.2 that the active learner performs differently on all the four datasets with different initial training sets, L1, L2 and L3. Similar results can be found in Figure 7.3 which shows the performance of the active learning process on the four datasets when the KMeans+ME method is used to select the initial training set. Figure 7.2 and Figure 7.3 show first that the non-deterministic clustering algorithms can produce very different clusterings even when applied to the very same dataset, and second that different initial training sets can have a significant impact on the outcome of the active learning process.



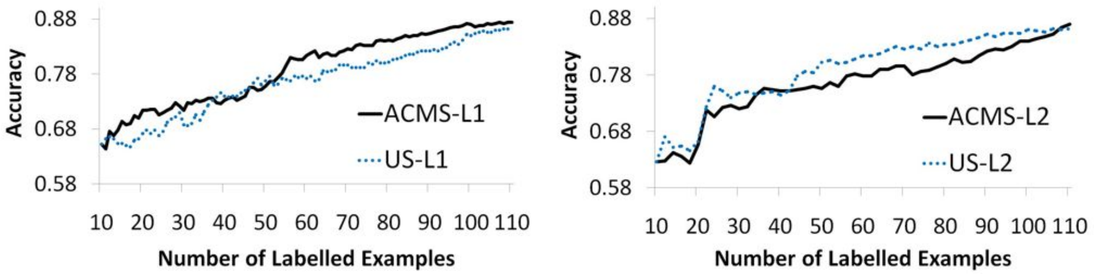
(a) WinXwin dataset

(b) Comp dataset

(c) Reuters dataset

(d) RCV1 dataset

Figure 7.3: The learning curves produced by three runs of the active learning process when the initial training set is populated using KMeans+ME clustering. Axes are zoomed for resolution.



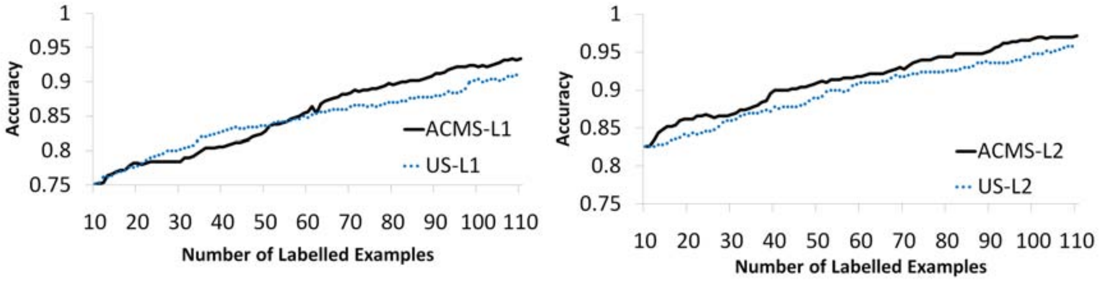
(a) Initial training set I

(b) Initial training set II

Figure 7.4: Comparison of the ACMS and US selection strategies on the Comp dataset using k-Medoids clustering created initial training sets.

This lack of determinism is especially damaging when comparing the performance of different selection strategies within the active learning process. Figure 7.4 and Figure 7.5 illustrate this problem. Here, two selection strategies, *uncertainty sampling* (US) and *aggregated confidence measures sampling* (ACMS) (details of which can be found in Hu *et al.* (2009) and Chapter 5), are compared. Figure 7.4 shows the comparison results on the Comp dataset with two initial training sets populated by k-Medoids clustering. Observations in Figure 7.4 show that the conclusion about whether ACMS is better than US can be different depending on which initial training set is used. Figure 7.4a shows that ACMS is better than US. A totally opposite conclusion can be made from Figure 7.4b which shows that ACMS is worse than US. This finding is supported by the results for two different runs of the experiment on the WinXwin dataset using KMeans+ME clustering to select the initial training set as shown in Figure 7.5 from which it is clear that due to the slightly different initial training sets selected each time it is very difficult to decide which selection strategy, if either, is performing better.

Clustering algorithms have been used to generate better initial training sets



(a) Initial training set I

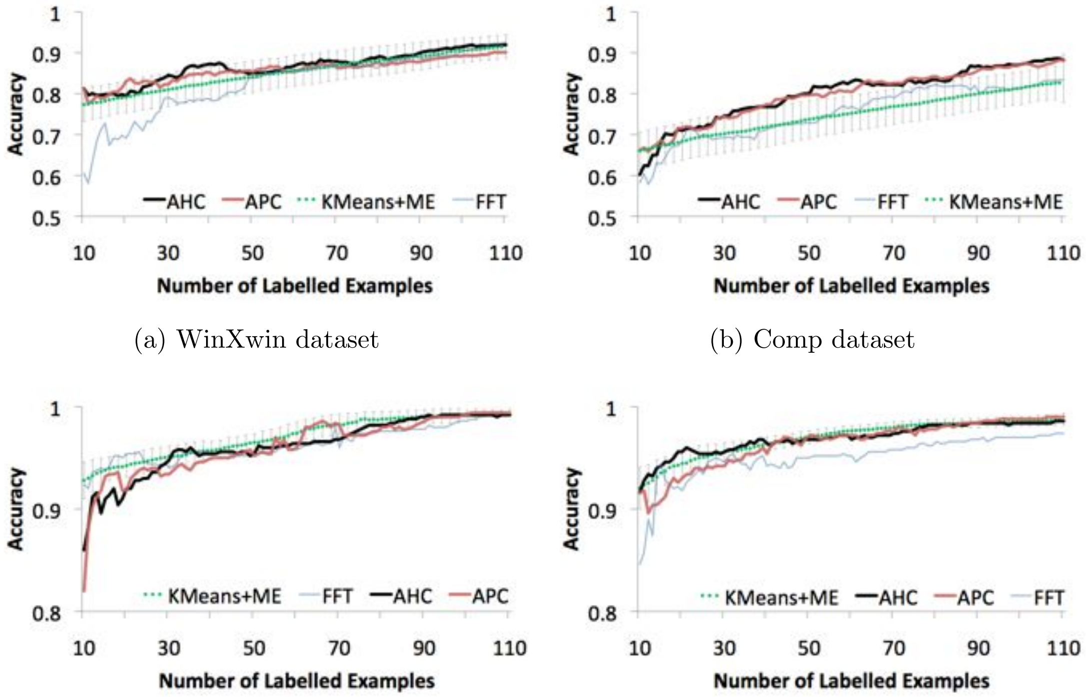
(b) Initial training set II

Figure 7.5: Comparison of the ACMS and US selection strategies on the WinXwin dataset using KMeans+ME created initial training sets.

for active learning than random sampling. However, existing methods use non-deterministic clustering algorithms. Experimental results show that non-deterministic clustering can result in inconsistent behaviour in the active learning process. In the next section, we propose novel initial training set construction methods, three deterministic clustering algorithms that have not been considered in the context of initial training set selection before.

7.2.3 Comparison of Different Clustering Techniques

The third group of experiments compare the performance of deterministic clustering techniques Furthest-First-Traversal, Agglomerative Hierarchical Clustering and Affinity Propagation Clustering with KMeans+ME (the best non-deterministic clustering method confirmed in Section 7.2.1) to see if a deterministic approach could manage comparable or better performance than the best non-deterministic one. Furthest-First-Traversal (FFT) is a very simple and computationally efficient way of deterministic clustering. Agglomerative Hierarchical Clustering (AHC) is a popular deterministic clustering algorithm which has been used in active learning. Affinity



(a) WinXwin dataset

(b) Comp dataset

(c) Reuters dataset

(d) RCV1 dataset

Figure 7.6: The learning curves produced by the active learning process when the initial training set is chosen using furthest-first-traversal (FFT), agglomerative hierarchical clustering (AHC), affinity propagation clustering (APC), random selection (RS), and KMeans+ME. Axes are zoomed for resolution. Legend order reflects order of curves based on AULC scores.

Propagation Clustering (APC) is a relatively new clustering algorithm which has shown good performance in various applications including text classification.

Figure 7.6 shows a comparison of FFT, AHC, APC and KMeans+ME algorithm (the best non-deterministic approach) in the active learning process on the four datasets. As KMeans+ME includes an element of randomness, the standard deviation error bars are shown to indicate the variation in the different 15 runs of the process used to calculate this average. The first observation from these results is that the FFT algorithm is not well suited to this task. This is not unexpected since, by choosing examples that are furthest away from each other, this algorithm

Table 7.2: AULC scores of active learning using different deterministic clustering algorithms in initial training set selection.

	WinXwin	Comp	Reuters	RCV1	Average Rank
FFT	82.6(5)	75.5(4)	96.4(2)	95.1(4)	3.75
AHC	86.7(1)	80.1(1)	96.2(3.5)	96.9(2)	1.88
APC	85.7(2)	79.9(2)	96.2(3.5)	96.5(3)	2.63
KMeans+ME	85.2(3)	75.1(5)	97.1(1)	97.0(1)	2.5
RS	84.6(4)	76.1(3)	95.9(5)	94.1(5)	4.25

is particularly susceptible to noise and outliers. The second observation is that the AHC and APC algorithms perform comparably to KMeans+ME, and on the Comp dataset both clearly dominate the non-deterministic technique which we have already indicated does not work well on this dataset.

Table 7.2 shows the AULC scores of the compared methods in populating initial training sets for active learning. It can be seen from the average ranks that AHC is better than KMeans+ME while APC is worse than KMeans+ME. For FFT, it is worse than random sampling on two out of the four datasets (WinXwin and Comp). As discussed before, this is because that FFT is prone to select outliers. For APC, it does not perform better than AHC. One possible reason is its limitations as discussed in Wang *et al.* (2008). The first limitation is the difficulty to set the optimal parameter for 'preference'. The second limitation is that it is hard to eliminate oscillations if occur. Another possible reason is that the implementation of AHC we employed was the best one found in Greene (2006). Across all the four datasets, all clustering algorithms compare very favourably to random sampling. Since AHC gives the best performance and it is deterministic, this makes it a better solution for selecting the initial training set for the active learning process.

7.3 Conclusions

Initial training set selection methods are investigated in this chapter. Most of the work in active learning uses randomly selected training examples to seed the active learning process. It has been previously established that using clustering to populate the initial training set can improve the performance of active learning systems. The commonly used clustering techniques for this task (k-Means, KMeans+ME and k-Medoids) are compared with a baseline approach which randomly picks examples to generate the initial training set. Our evaluation on a variety of textual datasets confirmed that using clustering to select the most representative examples to form the initial training set can help to improve the performance of the active learning process. However, these commonly used methods are non-deterministic which is problematic in its own right, and causes inconsistent performance when different active learning approaches are compared.

After demonstrating the problems caused by using non-deterministic clustering approaches, we examined the use of three deterministic techniques for populating the initial training set in the active learning process to fix the problem with non-deterministic clustering methods. Our experiments results show that better labelling accuracy to that achieved using the best of the non-deterministic approaches, can be achieved using one deterministic clustering algorithm – agglomerative hierarchical clustering. This better performance, and the determinism of agglomerative hierarchical clustering, clearly indicate that it is a better solution for selecting the initial training set for the active learning process.

Chapter 8

Reusability in Active Learning

In our previous work, if in the selection strategy one classifier is used (such as a k-NN classifier in ACMS), the same type of classifier is built on the labelled set selected by the active learning process to classify examples remaining in the unlabelled pool, i.e., labelled examples are reused by same type of classifier. This is known as the self-reuse scenario as discussed in Section 3.7.

However, in a wider domain, the classifier used in the active learning selection might be different from the classifier to be built on the labelled set achieved from the active learning process as discussed in Section 3.7. One example can be found in the heterogeneous uncertainty sampling (Lewis & Catlett, 1994) where a C4.5 induction program needs to be trained to produce decision rules on a set of labelled documents. Since it is impractical to use the C4.5 algorithm in the active learning process to generate the labelled documents, a more efficient probabilistic classifier was used in the active learning selection instead. This is one of the *reuse* scenario where an efficient classifier is required in the active learning selection to build a labelled set to train another, more expensive classifier. Another scenario is that when labelling a dataset using active learning, the classifier to be trained on this labelled set is unknown when labelling. One example can be found in a *crowdsourcing*[1] task. Crowdsourcing is the act of outsourcing a task traditionally performed by an employee or contractor to an undefined, generally large group of people in the form of an open call. An organiser might start an open call of a crowsourcing task about labelling a large collection of documents with their topics. When the participants label the dataset, they might not know the future use of the labelled set and they might pick the technique they prefer. So, some of them might use a k-NN based active learner to label the dataset. Others might use a Naïve Bayes classifier based active learner. After receiving the labelled set from the participants, the organiser might want to build an SVM classifier on it or he also could start another crowdsourcing task to build classification systems on the labelled set. In sample reuse scenarios, one problem need to consider is the resuability problem.

This chapter focuses on comparing the reusability of popular active learning methods for text classification. We investigate the problem of how well labelled training sets created using an active learning process can be reused by different classifiers. Futhermore, the computational performance of different active learning methods is also analysed. The analysis of reusability as explored in this chapter will shed some light on appealing applications of heterogeneous active learning ap-

[1] http://en.wikipedia.org/wiki/Crowdsourcing

proaches.

The structure of this chapter is as follows. Section 8.1 describes the evaluation first, detailing the data and the selection strategies based on three classifiers used to evaluate reusability, followed by the results of identifying the best classifier used to select examples for labelling in Section 8.2 and the best classifier to reuse examples selected using active learning in Section 8.3. Section 8.4 discusses the efficiency of active learning methods based on the three classifiers evolved in this research. Finally, Section 8.5 concludes with what we have learned regarding reusability.

8.1 Evaluation Methodology

The purpose of the evaluation is to compare the reusability of uncertainty based selection strategies in a pool-based active learning scenario using a set of popular classifiers for text classification.

8.1.1 Evaluation Objectives

There are two objectives to the evaluations described here. The first is to identify the best classifier (selector) used to select examples to label by active learning, i.e. to answer the question that when using active learning to generate a labelled set, what is the best classifier to use for text classification. The second is to identify the best classifier (consumer) to be trained on the labelled training set which has been obtained using active learning methods, or to answer the question that given a set of labelled documents derived from an active learning process, what classifier will perform best on this training set.

More precisely, the following four questions are to be answered:

Q1: If the consumer is known, which selector should be used?

Q2: If the consumer is unknown, which selector should be used?

Q3: If the selector is known, which consumer should be used?

Q4: If the selector is unknown, which consumer should be used?

When trying to identify the best selector, i.e. to answer Q1 and Q2, the terms *self-selection* and *foreign-selection* following Tomanek & Morik (2010) are used. Self-selection refers to the scenario where the selector and consumer use the same type of classifier. In contrast, foreign-selection specifies a scenario where the classifier used as the selector and the classifier used as the consumer are different. When trying to identify the best consumer i.e. to answer Q3 and Q4, we use the terms *self-reuse* and *foreign-reuse* which have the similar meaning to self-selection and foreign-selection but in situations of the availability of the selector instead.

8.1.2 Evaluation Set-up

This section describes the evaluation framework include datasets, selectors and consumers used in the experiments.

In order to conduct a comprehensive analysis, we tested various algorithms on seven textual datasets from the Reuters-21578[1], RCV1 (Lewis *et al.*, 2004), 20 Newsgroups[2] collections and a spam collection (Delany *et al.*, 2005b). As previously, each

[1] http://www.daviddlewis.com/resources/testcollections/reuters21578/

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

Table 8.1: Details of datasets used in the reusability evaluation experiments.

Dataset	**# of Examples**	**Pos. lbls**	**# of Features**	**# of Features (DF)**
20NG-WinXwin	1945	49.67%	16633	6235
20NG-Comp	1943	50.54%	13808	5278
20NG-Talk	1427	44.01%	14160	6402
20NG-Vehicle	1984	49.90%	14470	6698
Reuters-1804	1804	39.86%	7404	2327
RCV1-2000	2000	50.00%	10928	5877
Spam-1000	1000	50.00%	29985	6279

document is tokenised, stemmed using Porter Stemming and stopwords are removed. Each document is represented as a vector of words where the feature value represent the frequency of occurrence of the word. In addition, document frequency reduction is used where words that occur in less than three documents are removed. The properties of each dataset are shown in Table 8.1.

For each dataset, 1000 examples are initially randomly selected from the dataset. These selected examples are split into five equally sized, stratified folds to create five test sets. For each test set, five independent hold-out test sets of 200 examples are selected from the dataset, the remaining examples are used as the pool. Only the examples in the pool are used for the acquisition of the initial training set and the selection of active learning examples.

For each pool, as suggested in Hu *et al.* (2010c) and Chapter 7, an initial training set containing 10 seed examples is selected using agglomerative hierarchical clustering. 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 consumer's predictions are compared with the actual labels and the accuracy is recorded. This process is repeated until a label budget of 500 expires. The AULC score is then used to measure the overall performance. Averaged AULC results across the five pools for each dataset are reported.

The classifiers used in this study include a k-NN ($k = 5$) classifier using distance weighted majority voting; an SVM classifier with a linear kernel and a Multinomial Naïve Bayes classifier. The same configuration for each classifier is used, either it acts as the selector or the consumer in the active learning process. These three classifiers are the most common approaches used for text classification tasks. For non Naïve Bayes classifiers, i.e. the SVM classifier or the k-NN classifier, the term vectors are normalised to unit length.

The selection strategy used is uncertainty sampling. For the Naïve Bayes selector, as in Tomanek & Morik (2010), the ranking score of an example x is defined as in Equation 8.1 where the margin between the two classes defines the classification confidence. For the SVM selector, following previous work on active learning for SVM (Tong & Koller, 2001), the examples that are closest to the hyperplane are presented for labelling. The ranking score of x is defined in Equation 8.2 where $\Phi(x)$ is the feature vector of x and w_i is the SVM unit vector whose position is approximately in the center of the version space. For the k-NN selector, following our previous work (see Hu *et al.*, 2008, and Section 9.1), the ranking score $r(x)$ is defined as in Equation 9.3.

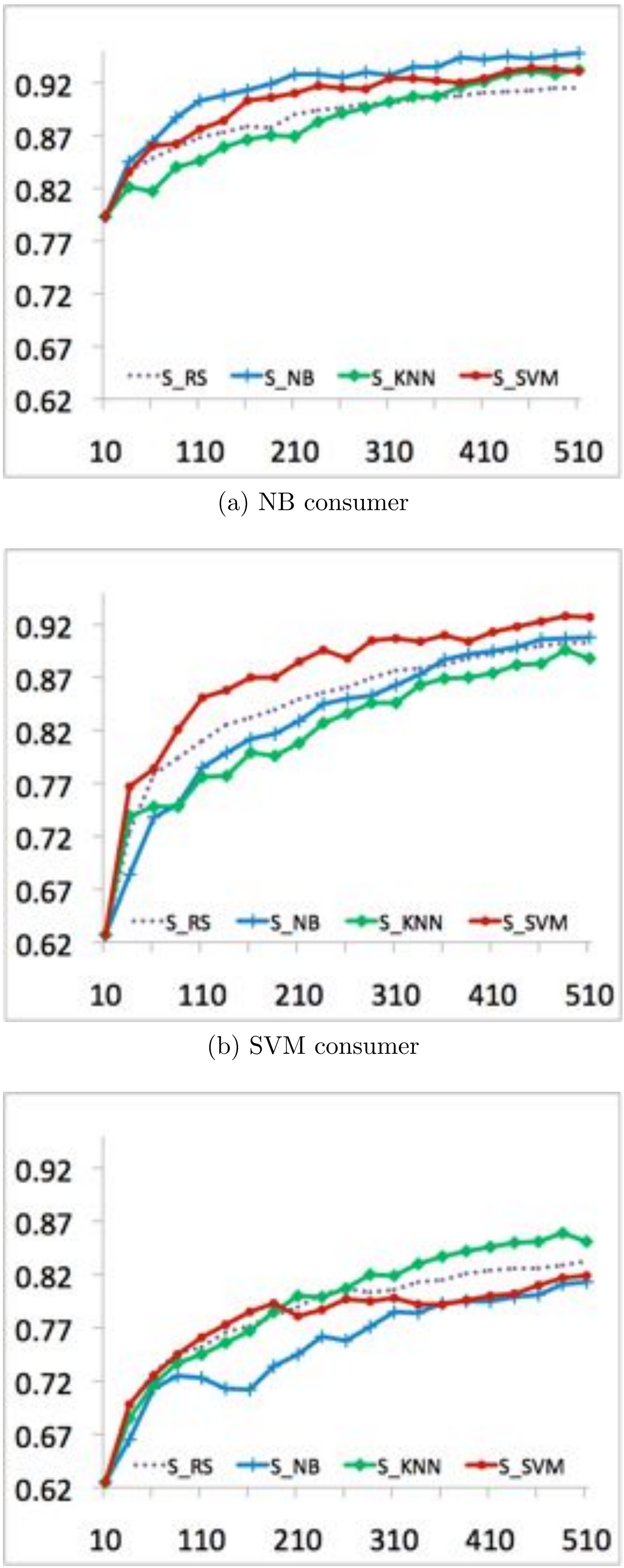
$$r(x) = 1 - |P(x \in class1) - P(x \in class2)| \tag{8.1}$$

$$r(x) = -|w_i \cdot \Phi(x)| \tag{8.2}$$

8.2 Identifying the Best Selector

There are two objectives to the identification of the best selector. The first is to compare the performance of self-selection, foreign-selection and random selection. The second and the more interesting aim is to find the best selector for text classification which tries to answer Q1 "which selector is the best if a known classifier will be used as the consumer in sample reuse" and Q2 "which selector is preferable if only unlabelled documents are available".

Figures 8.1 to 8.7 depict the learning curves for the NB, SVM and k-NN consumers respectively on the seven datasets. Note that to provide good legibility, the vertical axes do not range from 0.0 to 1.0. Here, for example, for the SVM consumer, the SVM selector (S_SVM) represents self-selection, while the NB selector (S_NB) and the k-NN selector (S_KNN) represent foreign-selection. We also include a random sampling selector (S_RS) which randomly selects examples to label as a baseline selector. We run the random sampling five times and report the averaged accuracy of the five runs. As can be seen from these figures, for both the NB and SVM consumers, the learning curve of self-selection dominates those of foreign-selection and random selection which indicates that self-selection performs better than foreign-selection and random selection. 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 confirmed one of the

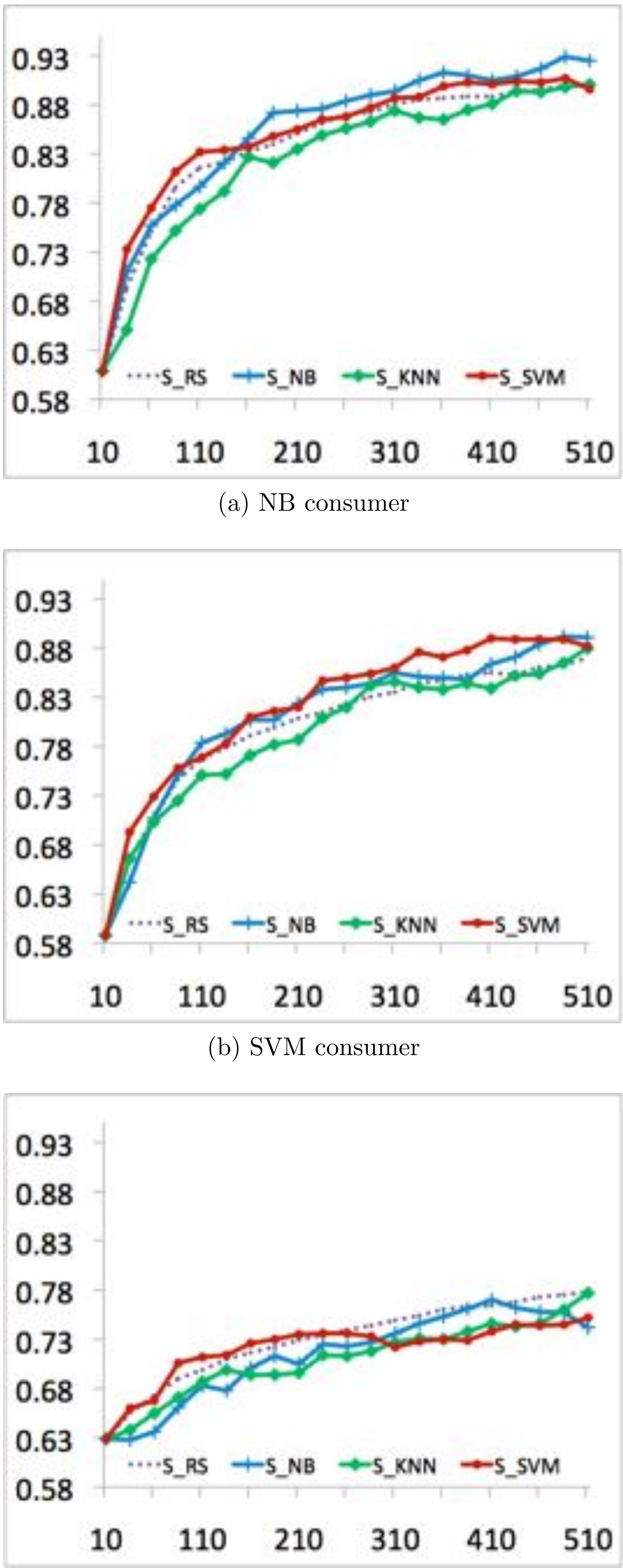


(a) NB consumer

(b) SVM consumer

(c) *k*-NN consumer

Figure 8.1: Learning curves of three consumers with different selectors (as shown in legend) on the 20NG-WinXwin dataset.

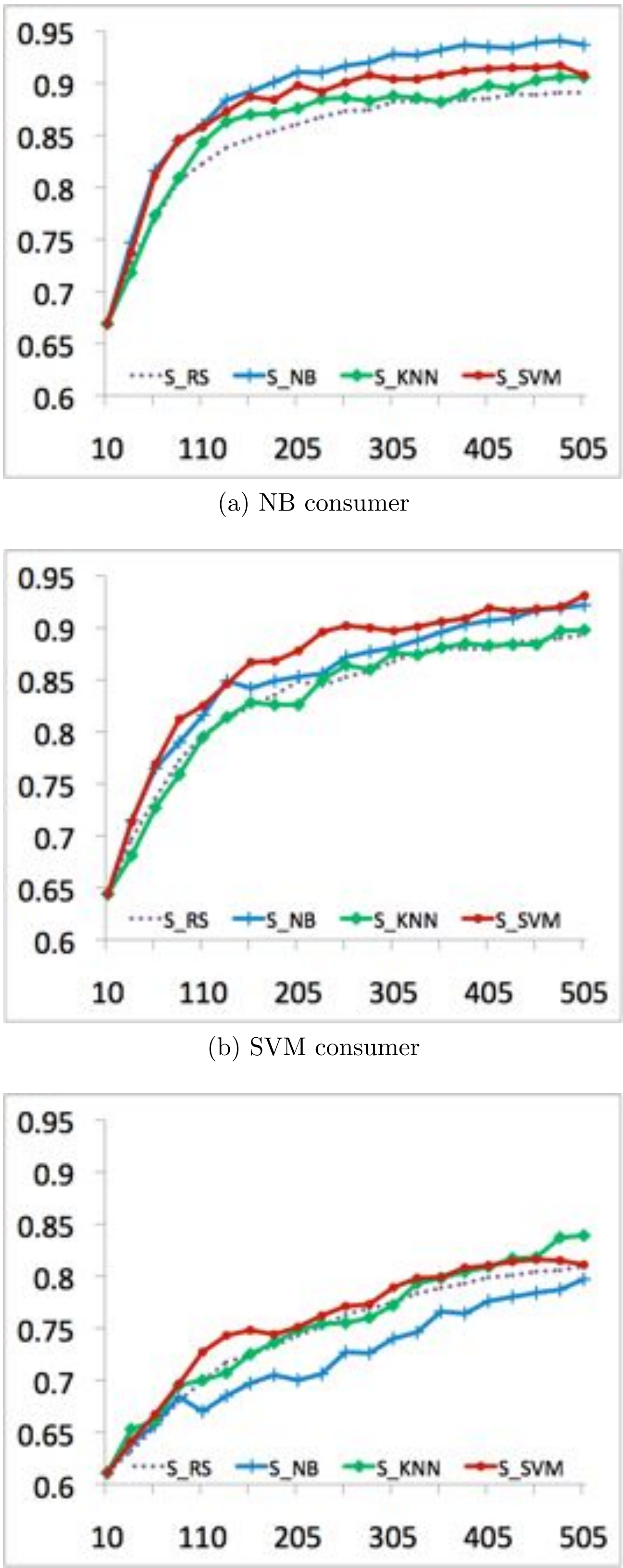


(a) NB consumer

(b) SVM consumer

(c) k-NN consumer

Figure 8.2: Learning curves of three consumers with different selectors (as shown in legend) on the 20NG-Comp dataset.

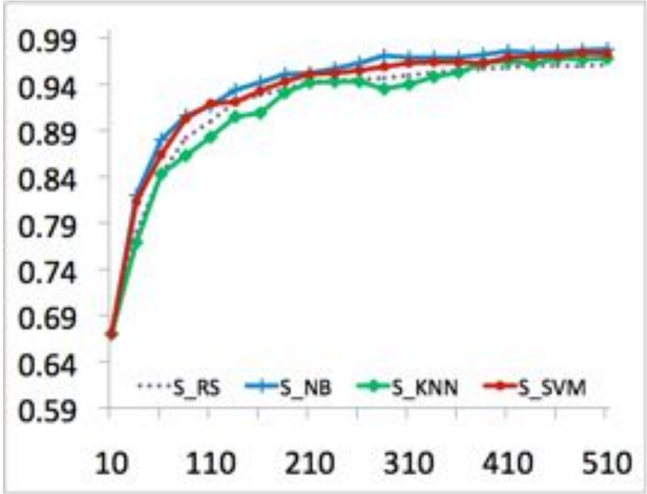


(a) NB consumer

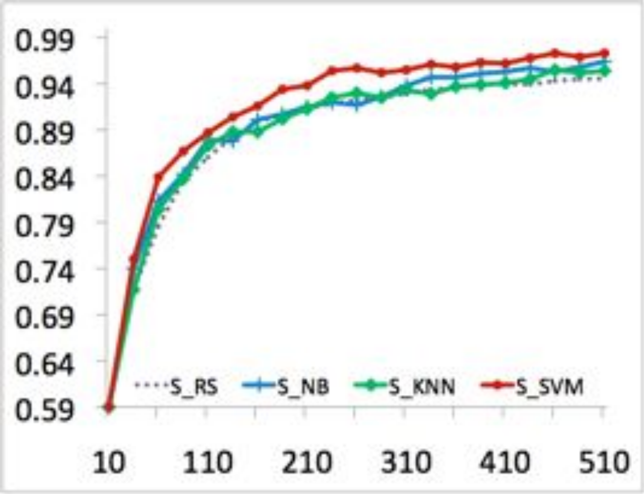
(b) SVM consumer

(c) k-NN consumer

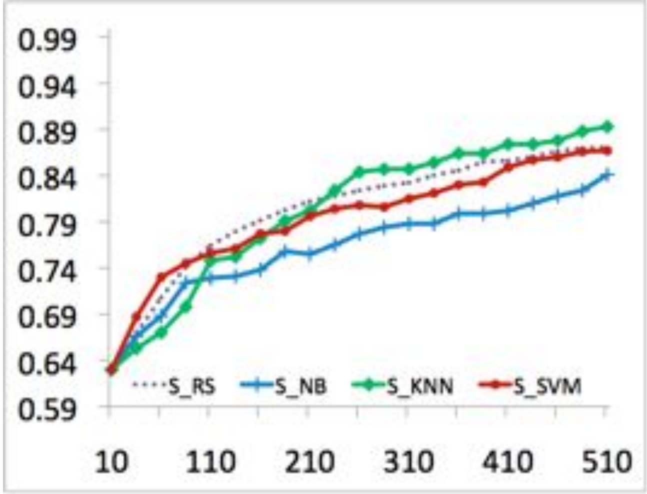
Figure 8.3: Learning curves of three consumers with different selectors (as shown in legend) on the 20NG-Talk dataset.



(a) NB consumer

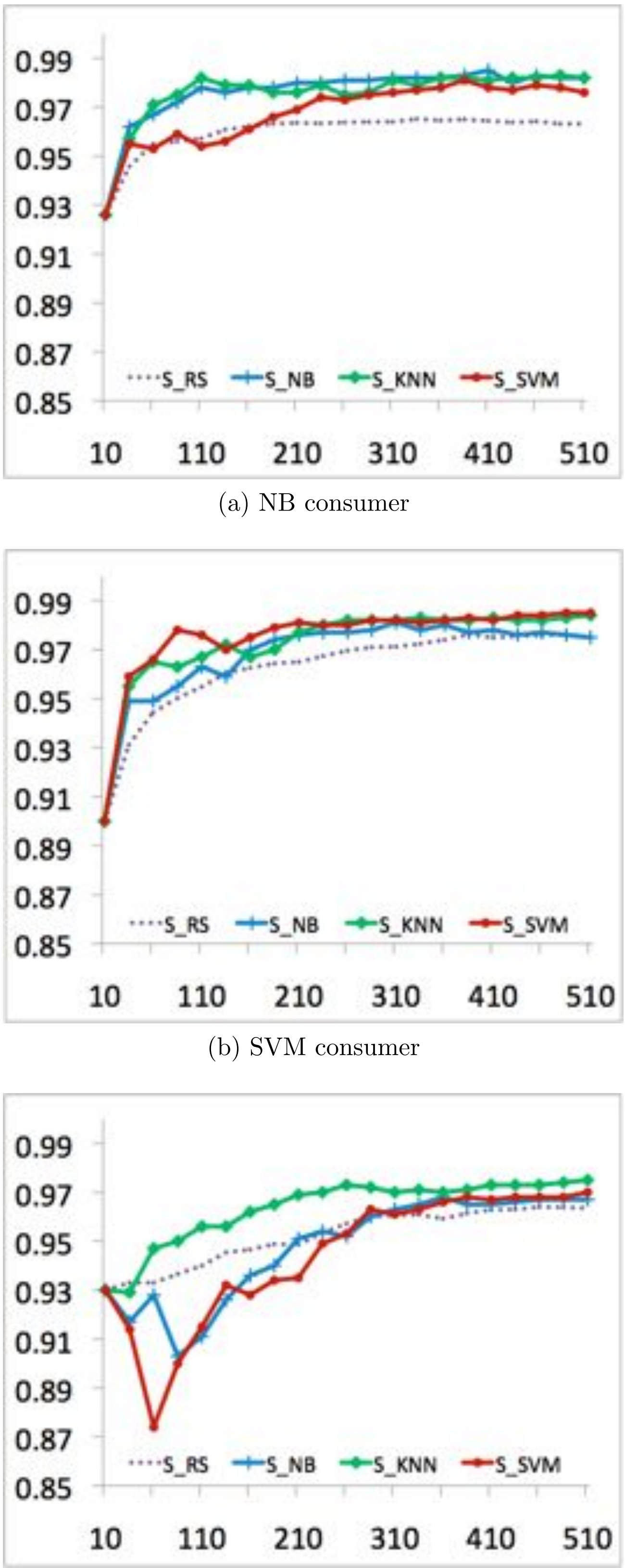


(b) SVM consumer



(c) *k*-NN consumer

Figure 8.4: Learning curves of three consumers with different selectors (as shown in legend) on the 20NG-Vehicle dataset.

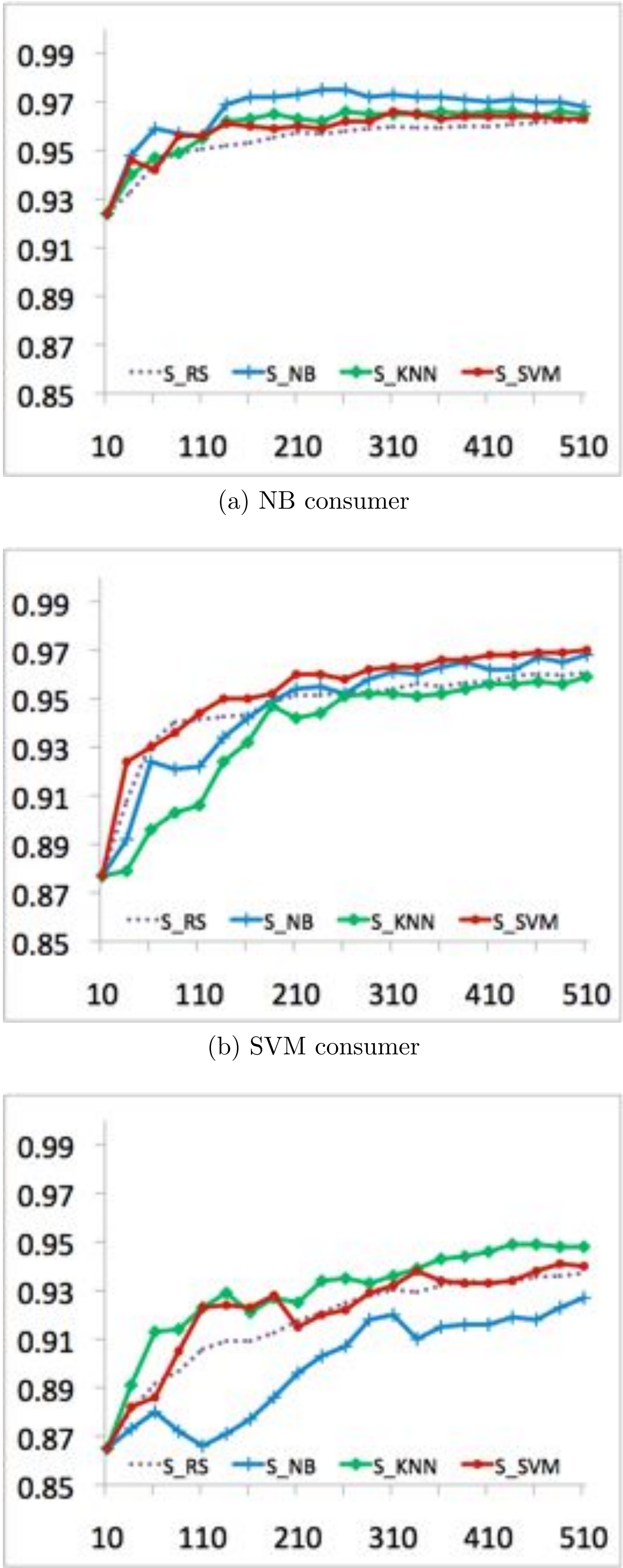


(a) NB consumer

(b) SVM consumer

(c) *k*-NN consumer

Figure 8.5: Learning curves of three consumers with different selectors (as shown in legend) on the Reuters-1804 dataset.

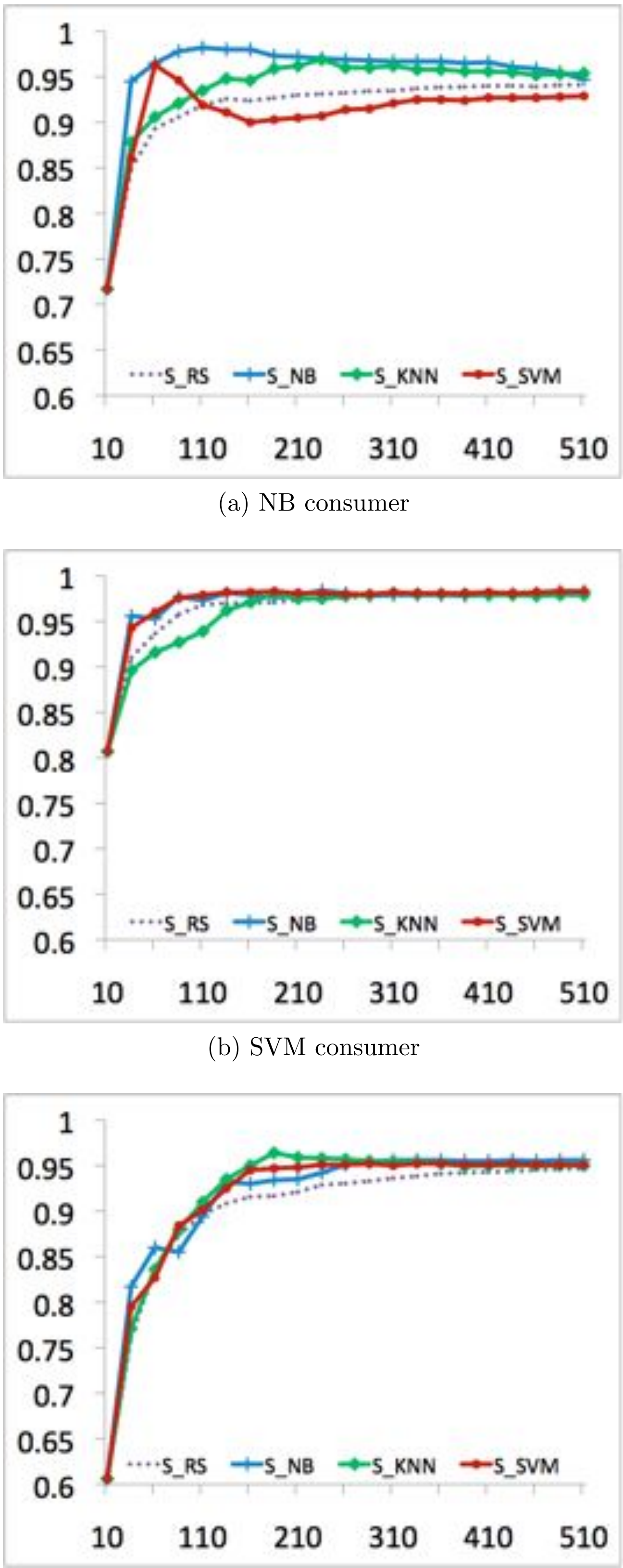


(a) NB consumer

(b) SVM consumer

(c) *k*-NN consumer

Figure 8.6: 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 8.7: Learning curves of three consumers with different selectors (as shown in legend) on the Spam-1000 dataset.

findings from Tomanek & Morik that self-selection is occasionally outperformed by foreign-selection (see R2 on Page 66).

In order to analyze the relationship between self-selection and foreign-selection in more detail, we calculated AULC scores. Table 8.2 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 and letters in parentheses (s, f, or r) indicate that the best selector is in a situation of self-selection, foreign-selection or random selection respectively.

From Table 8.2, it can be seen that when NB and SVM consumers are used, self-selection is better than both foreign-selection and random selection on all the seven datasets. When the k-NN consumer is used, results are 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 is the best one. On the 20NG-Talk and Spam-1000 datasets, SVM selector and NB selector are the best selectors respectively. Different from global classifiers such as the SVM and the NB classifier, the k-NN classifier is not well suited for self-selection. One possible reason is that the k-NN classifier is a local learner and has less bias to the examples used to train it. So it cannot gain as much benefit as global learners from the self-selected labelled training examples which makes self-selection not suited well here.

Overall it can be concluded from Table 8.2 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

Table 8.2: Results for identifying the best **selector**. Colors: **429.3**(s), best and selector and consumer match, i.e. self-selection is the best; **379.9**(f), best and selector and consumer do not match, i.e. foreign-selection is the best; **365.5**(r), random selection is the best; *441.4*, foreign-selection is worse than random selection.

		NB	KNN	SVM	RS
		Selector	**Selector**	**Selector**	**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**(r)
	20NG-Talk	*360.8*	376.8	**379.9**(f)	373.1
	20NG-Vehicle	*382.4*	403.0	*398.4*	**403.2**(r)
	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**

Q1 that self-selection should be used to pick the best selector if the consumer is known, i.e. the same type of classifier should be used in the selection to produce the labelled training data for a particular consumer.

From Table 8.2, it also can be seen that both NB selector and SVM selector performs best on 8 out of the 21 consumer/dataset pairs. In order to decide which one is better, average ranks are calculated across all the seven datasets and the three consumers. The average ranks for NB selector, k-NN selector, SVM selector and RS selector are 2.24, 2.83, 2.02 and 2.90 respectively. The results indicate that the best selector is the SVM selector, followed with the NB selector and the k-NN selector which is promising evidence that an SVM classifier generates training data that is most reusable. Q2 can then be answered that if the consumer to be used is unknown, the SVM classifier is the best choice to be used in the active learning process.

It is also interesting to notice the comparison results of foreign-selection and random selection. As shown in Table 8.2, cases where foreign-selection is worse than random selection are highlighted in italic font. Foreign-selection is worse than random selection on 4, 7 and 10 out of 14 foreign-selection scenarios when the NB, SVM and k-NN consumers are used respectively. The fact that for the NB consumer in most scenarios, foreign selection is better than random selection confirmed the finding R1 on Page 66 proposed in Tomanek & Morik (2010). For the scenarios where foreign selection is worse than random selection, among the 7 cases for the SVM consumer, 5 of them come from the scenarios when a k-NN selector is used in active learning while only 2 from an NB selector. The results indicate an SVM classifier and an NB classifier can mix well as the selector and the consumer but not very well with a k-NN classifier. This may be attributed to the fact that both the SVM and the Naïve Bayes classifier are eager-learners while the k-NN classifier is a lazy-learner. An informative example for an eager-learner may not be an informative example for a lazy-learner.

In order to check whether examples selected by an eager-learner and a lazy-learner are different, the overlap between the 500 examples selected by an SVM selector, an NB selector and a k-NN selector was calculated. Table 8.3 shows the overlap of L_{S_i} selected by the selector S_i and L_{S_j} selected by the selector S_j, denoted

Table 8.3: Percentage overlap of the labelled sets between different selectors.

Dataset	**OL(S_NB,S_SVM)**	**OL(S_NB,S_KNN)**	**OL(S_SVM,S_KNN)**
20NG-WinXwin	54.2% (1)	43.0% (2)	41.3% (3)
20NG-Comp	47.6% (1)	36.2% (2)	35.3% (3)
20NG-Talk	61.8% (1)	51.7% (2)	50.3% (3)
20NG-Vehicle	54.9% (1)	44.5% (2)	42.0% (3)
Reuters-1804	67.2% (1)	65.4% (2)	64.5% (3)
RCV1-2000	62.1% (1)	54.6% (3)	57.1% (2)
Spam-1000	65.1% (2)	60.8% (3)	69.8% (1)

by $OL(S_i, S_j)$. As can be seen from Table 8.3, there is a considerable difference in the labelled sets resulted from different selectors. The overlap of examples selected by the SVM selector and the NB 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 NB selector and the k-NN selector. This suggests that an eager-learner and a lazy-learner have different preferences for informative examples.

8.3 Identifying the Best consumer

The objective of this section is to identify the best consumer when trained with different sets of training examples produced by active learning.

The same results as in Section 8.2 are used, but illustrated in a slightly different way in order to indicate the best consumer more obviously as shown in Table 8.4. Similarly, if a classifier achieves the best performance on the labelled training examples selected by the same type of classifier in active learning, we say that self-reuse is the best sample reuse scenario as highlighting with a letter ‘s’ in the parentheses. Otherwise, if the best performance achieved from a labelled set generated by an active learning process using a classifier which is different from the consumer classifier, we say that foreign-reuse is the best sample reuse scenario as highlighting with a letter ‘f’ in the parentheses. For random selection, the best scores are highlighted in bold fonts.

The results show that self-reuse is not always the best reuse scenario which are consistent to the finding in Tomanek & Morik (2010) that examples selected by some type of classifier are not particularly better reused by the same type of classifier (see R3 on Page 66). Self-reuse is better than foreign-reuse only on 8 out of the 21 selector/dataset pairs while foreign-reuse is better than self-reuse on 13 out of the 21 selector/dataset pairs. For example, when the training examples are selected by an SVM classifier in the active learning process, foreign-reuse (using an NB consumer) is better on five out of the seven datasets.

Back to Q3, 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. One plausible reason is that the best classifier for one task not only depends on the training data used but also on the power of the classifier.

The last row of Table 8.4 shows the average rank of the three consumers across all the 28 selector/dataset combinations including the RS selector. It can be seen that overall the best consumer is the NB consumer, followed by the SVM consumer. This evidence answers Q4 that if the type of the selector is unknown, an NB classifier is the best choice to use as the consumer.

From the dataset point of view, the resulted AULC scores as same as in Table 8.2 and Table 8.4 are shown slightly differently in Table 8.5 where the best pair of selector and consumer is shown in bold text. It can be seen that on six out of seven

Table 8.4: Results for identifying the best **consumer**. Colors: **429.3**(s), best and selector and consumer match, i.e. self-reuse is the best; **365.5**, best and selector and consumer do not match, i.e. foreign-reuse is the best.

		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**	423.8	393.5
	20NG-Comp	**421.9**	402.0	365.5
	20NG-Talk	**425.4**	417.6	373.1
	20NG-Vehicle	**461.2**	446.7	403.2
	Reuters-1804	480.5	**482.0**	476.1
	RCV1-2000	**477.3**	473.2	459.1
	Spam-1000	460.8	**483.5**	454.0
Average Rank		**1.25**	**1.79**	**2.96**

datasets (except the Spam-1000 dataset), the best selector and consumer pair is the NB selector with the NB consumer. On the Spam-1000 dataset, the best selector and consumer pair is the SVM selector with the SVM consumer. The performance of the NB classifier on the Spam-1000 dataset is not as good as on other six datasets. One possible reason might be that the Spam-1000 dataset contains emails which are different from general articles, such as the length of the documents and the vocabulary used in the documents. Overall it seems clear that using the NB classifier in active learning selection and then built an NB classification model on the labelled examples offers the best performance on general text classification applications.

8.4 Efficiency of Selectors

To examine the efficiency of different selectors, experiments were conducted to compare the execution time for different selectors when selecting the same number of examples. Experiments are conducted on a MAC with Mac OS X Version 10.6.6 operation system, Intel® Xeon® CPU i3@3.06GHz with 4.0 GB 1333 MHz DDR3 RAM. For each dataset, all selectors are evaluated with an experiment of selecting 500 examples for labelling. Every experiment is repeated three times and average performance is reported.

Table 8.6 shows the time in seconds taken to select 500 examples on the seven datasets. From the results, it can be observed that among the compared selectors, the NB selector is the most efficient one due to the fact that the training and re-training process of the NB classifier is particularly efficient as discussed in Section 3.3.1.1. The k-NN selector is the second most efficient selector as the repeated

Table 8.5: Best **selector** and **consumer** pairs. Color: **429.3**, overall best combination.

	NB Consumer				SVM Consumer				*k*-NN Consumer			
	NB Selector	*k*-NN Selector	SVM Selector	RS Selector	NB Selector	*k*-NN Selector	SVM Selector	RS Selector	NB Selector	*k*-NN Selector	SVM Selector	RS Selector
20NG-WinXwin	**458.5**	441.4	451.3	443.7	416.6	409.9	437.9	423.8	377.6	397.9	389.5	393.5
20NG-Comp	**429.3**	413.6	426.6	421.9	407.8	397.1	413.1	402.0	357.7	355.1	360.5	365.5
20NG-Talk	**447.3**	430.6	439.5	425.4	427.2	417.1	434.0	417.6	360.8	376.8	379.9	373.1
20NG-Vehicle	**470.3**	459.2	467.1	461.2	451.0	447.8	460.5	446.7	382.4	403.0	398.4	403.2
Reuters-1804	**488.8**	488.3	484.2	480.5	484.9	487.4	488.7	482.0	473.8	481.9	471.8	476.1
RCV1-2000	**483.8**	480.1	479.4	477.3	473.0	467.6	476.9	473.2	449.5	465.5	461.0	459.1
Spam-1000	481.5	472.0	458.6	460.8	487.0	481.4	**487.5**	483.5	462.4	462.0	462.0	454.0

retraining required in active learning is especially efficient – new examples are simply added to the labelled set. The SVM selector is the least efficient one with a significantly longer execution time since in every learning iteration, the SVM model needs to be retrained and the training time increases with the number of examples in the training set.

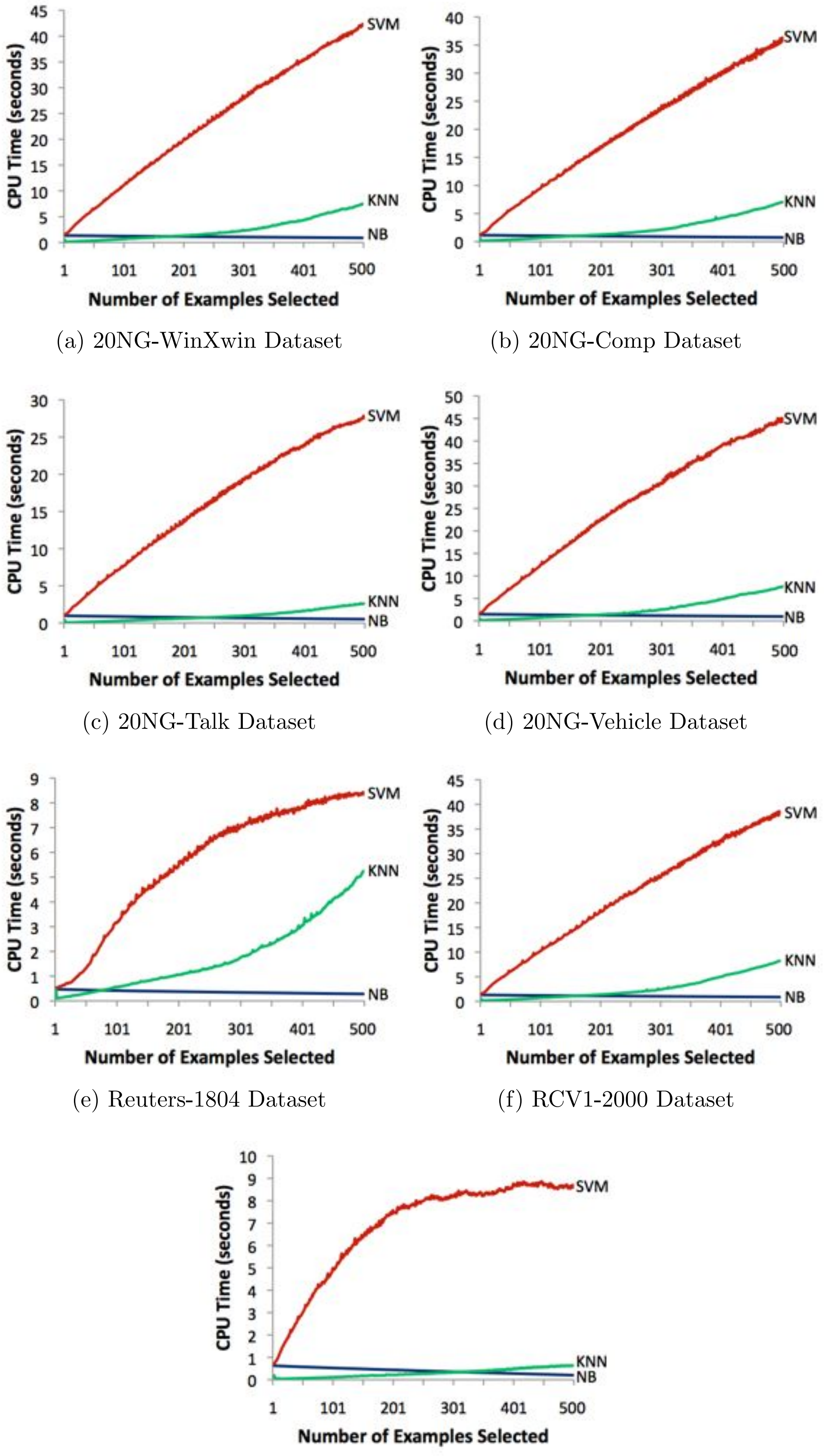
The graph of CPU time of the three selectors on the seven datasets is shown in Figure 8.8. It indicates the CPU time of the SVM selector increasing as the size of the labelled set grows. There is also a small increase for the k-NN selector while the CPU time for the NB selector remains stable and doesn't increase when the number of labelled examples increased. This suggests that if performance in terms of speed is a key factor, such as in scenarios of online learning, an NB or a k-NN classifier may be a better choice as selectors although an SVM classifier may have better performance in terms of reusability.

Table 8.6: Time performance on the seven datasets (seconds).

Dataset	**NB Selector**	**SVM Selector**	**k-NN Selector**
20NG-WinXwin	553	11659	1235
20NG-Comp	468	9895	1172
20NG-Talk	364	7988	488
20NG-Vehicle	611	12805	1322
Reuters-1804	183	2806	892
RCV1-2000	536	10683	1341
Spam-1000	201	3429	148

8.5 Conclusions

Understanding the reusability of training examples generated using an active learning selection strategy that uses a specific classifier is an interesting problem and



(a) 20NG-WinXwin Dataset

(b) 20NG-Comp Dataset

(c) 20NG-Talk Dataset

(d) 20NG-Vehicle Dataset

(e) Reuters-1804 Dataset

(f) RCV1-2000 Dataset

(g) Spam-1000 Dataset

Figure 8.8: CPU time of three selectors on seven datasets. Axes are zoomed for resolution.

can be very useful in practical applications. We compared the reusability of selection strategies based on three common classifiers for text classification and tried to answer the following questions:

- If we want to build a training set using active learning, which classifier is the best choice to use in the active learning selection?
- If a set of training examples is available which have been labelled using active learning, which classifier can perform best on it?

We have empirically studied the case of sample reuse and reusability on text classification problems and found that labelled training sets generated by active learning methods can be reused by multiple classifiers. The following conclusions can be made from our experiments:

- When using active learning to generate training examples for a particular classifier, it is better to use the same type of classifier in the selection than using a different one. This is confirmed when an SVM classifier, a Naïve Bayes classifier and a k-NN classifier are built on examples selected using active learning.
- Overall, the best classifier used in active learning selection regards reusability is the SVM classifier which confirms that SVM-based active learning method is very powerful in the text classification domain.
- Local and global classifiers don't mix well for 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. The use of the same classifier in the reuse as in the selection doesn't guarantee the best performance.

- Overall, the best classifier to reuse training 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 start to pick as the classifier to reuse the labelled training set. 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, SVM-based active learning selection strategy is the best choice for constructing the labelled training set in order to get better 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.

Chapter 9

Case Studies

During the work of this thesis, opportunities arose to explore using the techniques developed in this thesis in applications of using active learning in domains using structured textual data and multi non-text domains. The first case study is a prototype application to label recipe datasets (Hu *et al.*, 2008; Zhang *et al.*, 2008). The second case study is our experience in labelling unlabelled data sets from various domains provided by the 2010 Active Learning Challenge (Guyon *et al.*, 2010).

While this work does not contribute directly to the main claims in the thesis, they are interesting case studies in the applications of active learning. So they are included in this chapter.

9.1 Case Study 1: The Computer Cooking Contest

This section describes an initial prototype which was developed and used for an initial evaluation.

9.1.1 Problem Setting and System Configuration

A set of recipes from the 1st Computer Cooking Contest held at ECCBR'08 is used for the initial evaluation in which recipes are stored in an XML file. 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 collection of XML recipes was provided as part of the challenge, it did not include labels to indicate cuisine types or course types.

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 (Zhang *et al.*, 2008). 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 to evaluate the accuracy of the labels created by the labelling system.

Table 9.1 shows a few recipes which have been pre-processed into recipe titles and ingredient parts. Each recipe has a list of ingredients. WordNet (Fellbaum, 1998) is used to identify the food products from the ingredient text. Each food product stored for an ingredient is a food concept from WordNet. Labels are added to the recipe to indicate the recipe type.

The similarity measure used to compare two recipes in the k-NN classifier was based on a weighted combination of the similarity between the ingredients used in the two recipes, and the similarity between their titles. Equal weights of 0.5 for the ingredient similarity and 0.5 for the title similarity were used as shown in Equation 9.1. The title similarity measures the ratio of the number of same terms in both titles divided by the minimum number of terms in the two title strings of the recipes (as shown in Equation 9.2). When calculating the ingredient similarity ($sim(r_i.ings, r_j.ings)$) between recipe r_i and r_j, the recipe ingredient in recipe r_i that best matches ingredients in recipe r_j ($r_j.int_1, r_j.int_2, \ldots, r_j.int_m$) is found for each ingredient $r_j.int_k$. This is done by calculating pair-wise similarities between any two ingredients from the two recipes and the one has the highest similarity is the best match. The similarity between two ingredients is measured based on the WordNet ontology. The main food concept is parsed from the ingredient's textual description and matched to a valid concept in WordNet. Then the semantic relatedness of ingredients is computed according to the method described by Jiang & Conrath (1997). The overall ingredient similarity for recipe r_i and recipe r_j is then calculated as a weighted sum of the similarities between each ingredient in recipe r_j and its appropriate best match ingredient in recipe r_i. Further details of this can be found

Table 9.1: Recipe structures.

Recipe Label	Recipe Title	Ingredient 1	Ingredient 2	Ingredient 3
Desserts	Apple Crumble Pie	pie shell	apples	sugar
Main Dish	Mexican Pie	onion	pepper	flour
Desserts	Special Apple Pie	flour	apples	sugar
Main Dish	Saucy Bean'N Beef Pie	beef	onion	beans
Desserts	Upside down apple Pie	sugar	flour	apples
?	Shepherd's Pie	beef	onion	potatoes

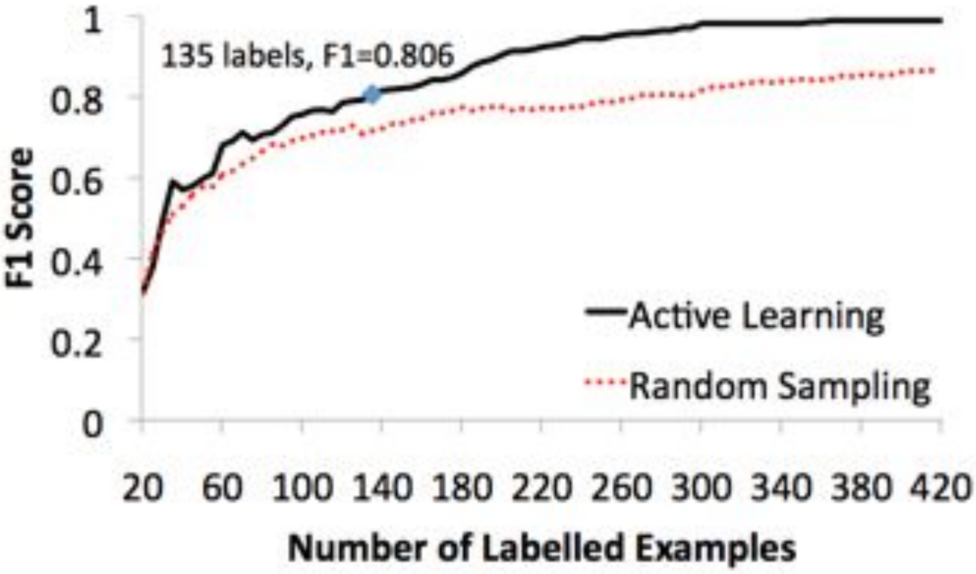


Figure 9.1: F1 score comparison of active learning and random sampling.

in Zhang *et al.* (2008).

$$sim(r_i, r_j) = 0.5sim(r_i.title, r_j.title) + 0.5sim(r_i.ings, r_j.ings) \tag{9.1}$$

$$sim(r_i.title, r_j.title) = \frac{|r_i.title \cap r_j.title|}{\min(|r_i.title|, |r_j.title|)} \tag{9.2}$$

In this initial prototype, initial training examples are randomly selected. The selection strategy used is an uncertainty sampling approach using a k-NN classifier ($k = 5$). Our choice of the k-NN classifier is informed by the fact that we feel it is

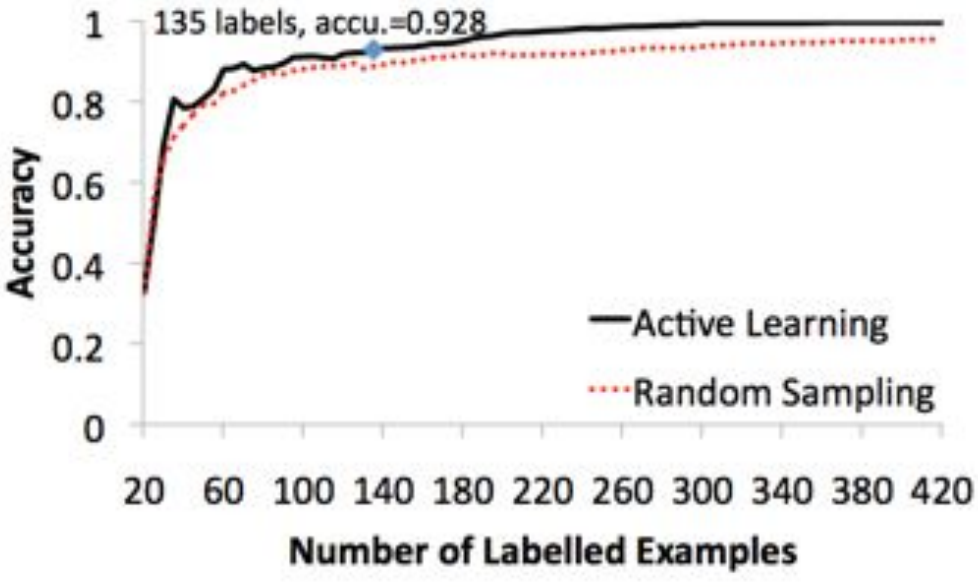


Figure 9.2: Accuracy comparison of active learning and random sampling.

uniquely suited to the active learning labelling task as discussed in Section 3.3.1.1 on page 45. In the selection strategy, the pool examples are ranked according to the ranking score $r(x)$ as defined in Equation 9.3 where $NN_i(x)$ is the i_th nearest neighbour of example x and $k = 5$. The example with the highest ranking score is selected for labelling first. As discussed in Section 3.3.1, the motivation for this is that this is the example that the system is currently most unsure about, and so labelling it will result in the most benefit to the system.

$$r(x) = -|\sum_{\substack{NN_i(x)\in class1\\1\le i\le k}} Sim(x, NN_i(x)) - \sum_{\substack{NN_i(x)\in class2\\1\le i\le k}} Sim(x, NN_i(x))| \tag{9.3}$$

9.1.2 Evaluation

The objective of the evaluation was to confirm that the ALL system can help to generate labelled datasets. 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 training set for the ALL system, described in section 4.1, which was allowed to run for a label-budget of 420 including the initial training set labelling. Since the recipe dataset is an unbalanced dataset in which 16% of the examples are from one class and the rest are from another class, both F1 scores and accuracies are used as the performance measures. After each labelling the accuracy of the labels applied to each recipe was calculated, as was the F1 score. In order to act as a comparison, a second experiment was run in which the examples to be labelled by the human expert were randomly selected rather than using the ranking score. Again, this was allowed to run to a total of 420 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 of human labels are shown in Figure 9.1, while the accuracies are shown in Figure 9.2.

As can be seen from Figures 9.1 and 9.2 the ALL system outperforms the system based on random sampling. In fact, after just 135 labels (approximately 15% of the entire dataset) the accuracy achieved by ALL has reached over 92.8%, with an associated F1 score over 0.8. It is also interesting to notice that it needs 95 labelled examples to achieve the accuracy of 91% while the classification accuracy on the whole dataset using 10-fold cross validation is 89.96% (averaged from two runs of 10-fold cross validation). The 95% confidence interval for ALL for the mean of accuracies is [0.9312, 0.9486] while for random sampling is [0.8877, 0.9039]. The 95% confidence interval for ALL for the mean of F1 scores is [0.8492, 0.8774] while for random sampling is [0.7423, 0.7626]. Paired t-tests show that the improvement of ALL in both F1 score and accuracy are statistically significant ($p < 0.0001$).

Although the ALL system in this initial evaluation is based only on simple active learning techniques, the experimental result shows that it can help to build high performance labelled dataset with much less labelling effort.

9.2 Case Study 2: The 2010 Active Learning Challenge

The 2010 Active Learning Challenge[1] (Guyon *et al.*, 2010) addressed problems where labeling examples are expensive, but unlabeled data are available at low cost and in large amount. Datasets from various domains (chemo-informatics, handwriting recognition, text processing, ecology, marketing, and embryology) were made available for the challenge. In this challenge, the participants were given an intial budget of virtual cash and were allowed to purchase labels until their cash was used up . The goal is to reach the best prediction performance as fast as possible. More details and results of the challenge can be found in Guyon *et al.* (2010). Since the datasets supplied come from different domains with different difficulty levels, such as different sparsity and different missing rate, we used SVM classifiers in our participation of the active learning challenge. Our methods and the results of the proposed methods are discussed in this chapter.

Active learning methods with support vector machines have gained wide accep-

[1] http://www.causality.inf.ethz.ch/activelearning.php

tance because of their significant success in numerous real-world learning tasks. As discussed in Chapter 3 (see Section 3.3.1), uncertainty sampling is commonly used to select the most informative examples to present to an oracle for labelling, which in the case of an SVM classifier are the examples closest to the decision hyperplane. Our initial experiments seems to indicate that combining exploration with uncertainty sampling improves performance on certain datasets but not all. This section proposes using exploration guided approaches to select both informative and representative examples to present for SVM-based active learning. We aim to investigate methods balancing exploitation with exploration in active learning to improve the performance of uncertainty sampling using SVMs in very large datasets.

Our selection approach complements SIMPLE MARGIN by exploring the examples chosen by SIMPLE MARGIN and selecting a representative sub-sample. This reduces the possibility of selecting duplicates and/or noisy examples. The first approach uses *Affinity Propagation Clustering* (APC) (Frey & Dueck, 2007) to cluster the N examples sampled by SIMPLE MARGIN and chooses the cluster centers to present for labelling. We call this approach *SIMPLE+APC*. The second method employs EGAL for the purpose of exploration. We call this method *SIMPLE+EGAL*.

We compare our augmented approaches to using SIMPLE MARGIN alone on six datasets from the 2010 Active Learning Challenge. The properties of the six datasets are shown in Table 9.2. Columns entitled “Features” and “Examples” show the number of features and examples in each dataset. Columns entitled “Sparsity” and “Missing” denotes the percentage of sparsity and missing data respectively. The column entitled “Pos. lbls” denotes the number of positive labels in the datasets.

Table 9.2: Datasets from the 2010 Active Learning Challenge.

Dataset	Domain	Featuers	Sparsity	Missing	Pos. lbls	Examples
IBN SINA	Handwriting	92	80.67%	0%	37.84%	10361
NOVA	Texts	16969	99.67%	0%	28.45%	9733
SYLVA	Ecology	216	77.88%	0%	6.15%	72626
ORANGE	Marketing	230	9.57%	65.46%	1.78%	25000
HIVA	Chemoinformatics	1617	90.88%	0%	3.52%	21339
DOCS	Texts	12000	99.67%	0%	25.52%	10000

The datasets are pre-processed to replace missing values with modes and means. The dataset is clustered first using affinity propagation clustering and cluster centres are selected for labelling to seed the active learning process.

The prediction performance is measured by *GlobalScore* which is a normalised score calculated from the area under the learning curve which plots the area under the ROC curve in y-axis, as a function of the number of labels queried in x-axis in a logarithm scale. More details about how to compute the GlobalScore can be found in Guyon *et al.* (2010).

At each iteration k examples are sampled for labelling. For SIMPLE MARGIN the k examples closest to the hyperplane are selected for labelling. For the hybrid approaches SIMPLE+APC and SIMPLE+EGAL, SIMPLE MARGIN selects N ($N \simeq 10 * k$) examples closest to the hyperplane first, then from which a subset of k examples are chosen for labelling by different exploration methods. The value of k was increased logarithmically in order to match with the scale of the evaluation measure.

As discussed in Section 3.3.3.1, in the later stage of the active learning process, exploitation should become the main concern. So, after eight iterations (more than 500 examples getting labelled) only exploitation based selection strategy – SIMPLE

MARGIN is used in active learning selection.

The overall performance figures with their ranks shown in parentheses are included in Table 9.3. It shows that both SIMPLE+APC and SIMPLE+EGAL outperform SIMPLE MARGIN alone on the IBN_SINA, NOVA, SYLVA and HIVA dataset. However on the ORANGE and HIVA dataset, all three approaches perform poorly with GlobalScores less than 0.5 which means that these active learning methods are worse than random sampling. One plausible reason is that both the ORANGE dataset and the HIVA dataset are very unbalanced datasets (with less then 4% examples from the positive class) which makes them very hard to label. The last row of Table 9.3 shows the average ranks of the three compared methods which show that both SIMPLE+APC and SIMPLE+EGAL outperform SIMPLE MARGIN across the six datasets.

Table 9.3: GlobalScores on the six datasets.

Dataset	SIMPLE MARGIN	SIMPLE+APC	SIMPLE+EGAL
IBN SINA	0.7793 (3)	0.8340 (1)	0.8189 (2)
NOVA	0.5036 (3)	0.5750 (2)	0.5846 (1)
SYLVA	0.8693 (3)	0.8833 (1)	0.8817 (2)
ORANGE	0.1516 (1)	0.1459 (3)	0.1460 (2)
HIVA	-0.0052 (3)	0.1324 (1)	0.0949 (2)
DOCS	0.6119 (2)	0.5910 (3)	0.6307 (1)
Avg. Rank	**2.50**	**1.83**	**1.67**

The results show that our approach, which enhances the SIMPLE MARGIN uncertainty sampling approach with an exploration guided technique, improves the performance of the standard SVM-based active learning method.

9.3 Conclusion

An application of using active learning on a set of recipes showed the usefulness of active learning in labelling structured textual datasets. Another example of using active learning on datasets from the 2010 Active Learning Challenge was described. Experiments on multiple domains other than text classification of using active learning show that augmenting uncertainty sampling with exploration-based methods can be beneficial.

Chapter 10
Conclusions

Supervised machine learning approaches are used extensively to solve real-world challenges. One core area of supervised learning approaches is text classification. However, the success or failure of classification systems depends on the datasets used to train them. Without a good dataset it is impossible to build a quality system. 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 classes. 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). 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. This thesis examines the applicability of active learning in such scenarios for the rapid and economical creation of labelled training data.

We started this work by designing an active learning labelling system for text classification applications and aimed to reduce overall labelling costs by actively querying the most informative examples. We targeted various problems associated with using active learning methods to label textual datasets.

As we pointed out in the introduction, there are several complex factors that have to be addressed for this task. This work has helped to answer outstanding questions about active learning in text classification applications, e.g., "how similarity-based active learning can help in active learning labelling" and "what methods are best for documents labelling tasks".

In this final chapter, we briefly summarise key contributions of this work and discuss several future work directions. The central claim of this thesis is that labelled dataset creation for text classification can be accomplished by using active learning. A suite of algorithms that aim to reduce the amount of labelled training data have been implemented and empirically tested to support this claim.

10.1 Summary of Contributions and Achievements

Throughout the thesis, we have made arguments of why this work is an important contribution to the active learning community. The goal of this section is to collect these arguments, at the end of the thesis, in order to create a more coherent and full picture of how this work contributes to active learning research. Specific contributions include:

- *A new selection strategy for active learning using k-NN based confidence measures* – Chapter 5 described a novel approach to active learning using an advanced aggregated confidence measurement instead of the direct output of classifiers to measure the confidence of the prediction and choose the examples with least confidence for querying. Typically in active learning algorithms the most informative examples are selected for labelling through uncertainty sampling based on classification scores. However, previous work has shown that, contrary to expectations, there is not a direct relationship between classification scores and classification confidence. To address this issue, we proposed a novel technique using advanced aggregated k-NN classifier confidence measures in an active learning selection strategy. Experimental results on various textual datasets showed that the performance of this strategy is better than one based on uncertainty sampling solely based on classification scores.

- *A simple but effective exploration-only selection strategy for active learning in the textual domain* – Chapter 6 presented an exploration-only selection strategy, namely EGAL, which uses only the notions of density and diversity, based on similarity, in its selection strategy. EGAL avoids the drawbacks associated with exploitation-based approaches to selection. EGAL strategy differs in that it attempts to select examples based on structured information to sample wider and interesting area of feature space. Comparative experiments showed that EGAL is computationally efficient and it performs better than traditional exploitation-based methods in the earlier labelling phases.

- *A demonstration of using visualisation to understand active learning selection strategies* – Section 6.4 demonstrates how spring model based visualisations can be used to provide insight into the precise operation of various selection strategies.

- *The use of deterministic clustering techniques for populating the initial training* *set in the active learning process* – Chapter 7 illustrated the problems associated with using non-deterministic clustering for initial training set selection for active learning and then investigated the use of deterministic clustering techniques to bootstrap the active learning process. The initial training set used to seed the active learning process plays an important role. Previous work has shown that using clustering algorithms, such as k-Means and k-Medoids, to select the initial training set can accelerate the active learning process. However, the clustering techniques typically used are nondeterministic which causes inconsistent behaviour in the active learning process. We first illustrated the problems associated with using non-deterministic clustering for the initial training set selection in active learning. We then examined the performance of three deterministic clustering techniques, furthest-first-traversal, agglomerative hierarchical clustering, and affinity propagation clustering, for this task. Our experimental results on a collection of text classification datasets showed that the performance using deterministic clustering techniques is comparable to that of the non-deterministic approaches and it can be achieved without variations in behaviour.

- *The identification of the best classifier to use in active learning for text classification considering the reusability* – Chapter 8 investigated the problem of how general are labelled training sets created using active learning. Most algorithms for active learning use a criterion based on a specific classifier. These methods are tuned to that particular classifier since a set of labelled examples which is most informative for one classifier is not necessarily as informative for another classifier, resulting in poor reusability when reusing the labelled data by other classifiers. It is desirable that the labelled dataset has good reusability regardless of the type of classifiers which will be trained using it. This is particularly so when active learning is used for dataset labelling instead of building classifiers. We compared popular active learning methods for text classification and their reusability: SVM based uncertainty sampling, Naïve Bayes based uncertainty sampling and k-NN based uncertainty sampling. Experiments results showed that the SVM based active learner is the best one with highest reusability and NB classifier can reuse examples selected by various active learners.

10.2 Open Problems and Future Work

The research in this thesis represents important steps in using active learning for labelling textual datasets. However, every solution naturally generates more questions. Therefore, this section introduces some of the research directions which are closely related to the work in this thesis and appear most promising:

- Extend the current active learning labelling framework to deal with multi-class and multi-label classification problems.

- Incorporate class information into the design of EGAL. At the moment, only the distance information among examples and the information about whether one example is labelled or not is used in the design of the EGAL algorithm. It would be interesting to incorporate the class information of the manually labelled examples to develop more powerful algorithms.

- Combine exploitation based methods with exploration based methods. We have shown that exploration based selection strategies are preferable in the initial learning stage and with for more labelled examples, exploitation based selection strategies are more powerful. A better choice would be a combined method of exploration with exploitation. EGAL has been combined with an SVM-based selection strategy. Further work could be done on combining EGAL with a k-NN-based selection strategy, such as ACMS.

- Expand the work on initial training set selection. We intend to further examine deterministic versions of the k-Means algorithm. KMeans+ME performs very well when compared to agglomerative hierarchical clustering and affinity propagation clustering but suffers from the fact that it is non-deterministic. Although there is no agreed best technique for doing so, it is possible to modify the KMeans+ME algorithm to perform deterministically.

- Design stopping criterion. The 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. More sophisticated stopping criteria could be designed.

- Incorporate visualisation methods. Visualisation techniques have been used to understand selection strategies. In order to produce better interactive learning experience, visualisation is necessary. Further research could be proposed to use visualisation to help the design of selection strategies. For example, visualisation techniques can be used to avoid selecting outliers in uncertainty sampling or to find the best switch point for ensemble based selection strategies.

- Explore more factors on reusability problem. Previous work has shown that both model relatedness and sample similarity can not explain reusability. It would be interesting to discover the supporting factors of reusability, i.e., what factors contribute to higher reusability.

Finally, it would be interesting to apply methods developed in this work to more real-life applications, such as drug discovery and bioinformatics since they also have high-dimentional data and difficulty in getting labelled data.

Appendix A

Notation

g: a map function $g : X \rightarrow Y$

x, x_i, e: an example

X: set of all examples

y: an output value

Y: set of all output values

$\mathbb{R}$: set of real numbers

C: set of classes

c_i: class i

$\vec{x}$: feature representation of example x, with $\vec{x} = (f_1(x), \ldots, f_k(x))$

t_i: term i

d_j: document j

$\vec{d_j}$: feature representation of document j

$\vec{w}$: a weight vector

$\mathcal{S}$: a selection strategy

$\mathcal{O}$: an oracle (a human expert)

$\mathcal{D}$: set of all examples

$\mathcal{L}$: set of labelled examples

$\mathcal{U}$: set of unlabelled examples

$\mathcal{SC}$: a stopping criterion

$NN_i(t)$: the ith nearest neighbour of example t

$NLN_i(t)$: the ith nearest *like* neighbour to example t

$NUN_i(t)$: the ith nearest *unlike* neighbour to example t

M_i: a confidence measure

$thres_{ij}$: a confidence threshold for a confidence measure M_i and a class j

N_i: a pre-defined neighbourhood of x_i

Appendix B

Abbreviations

k-NN	k-Nearest Neighbour	p. 19
AL	Active Learning	p. 32
ALL	Active Learning based Labelling	p. 78
APC	Affinity Propagation Clustering	p. 135
AHC	Agglomerative Hierarchical Clustering	p. 135
ACMS	Aggregated Confidence Measures Selection strategy	p. 95
AULC	Area Under the Learning Curve	p. 58
AUC	Area Under the receiver operating characteristic Curve	p. 28
AI	Artificial Intelligence	p. 1
BOW	Bag-Of-Words	p. 16
DF	Document Frequency	p. 17
ECOC	Error Correcting Output Codes	p. 23
EGAL	Exploration Guided Active Learning	p. 50

FFT	Furthest-First-Traversal	p. 135
GERM	Global Entropy Reduction Maximization	p. 70
HPSG	Head-driven Phrase Structure Grammar	p. 65
IDF	Inverse Document Frequency	p. 18
KFF	Kernel Farthest First	p. 54
LVO	Learning Vector Quantisation	p. 52
MTAL	Multi-Task Active Learning	p. 71
NB	Naïve Bayes	p. 24
NLP	Natural Language Processing	p. 72
NN	Nearest Neighbour	p. 19
NUN	Nearest Unlike Neighbor	p. 48
PRBEP	Precision-Recall BreakEven Point	p. 28
PCA	Principal Components Analysis	p. 60
QBC	Query-By-Committee	p. 41
RBF	Radial Basis Functions	p. 23
ROC	Receiver Operator Characteristic	p. 28
REU	REUsability score	p. 68
SUD	Sampling by Uncertainty and Density	p. 53
SOM	Self-Organizing Map	p. 60
SVM	Support Vector Machines	p. 21
TF	Term Frequency	p. 17
TDM	Term×Document Matrix	p. 16
TC	Text Classification	p. 14

TCMNN	Transductive Confidence Machine for Nearest Neighbours	p. 48
VOI	Value-Of-Information	p. 72
VSM	Vector Space Model	p. 16
WKNN	Weighted k-Nearest Neighbour	p. 21

Appendix C

Additional Material for Chapter 7

C.1 Clustering

Clustering is an unsupervised learning method which groups together examples that are similar to each other into a *cluster*. Several different variants of an abstract clustering problem exist. A flat (or partitional) clustering produces a single partition of a set of objects into disjoint groups whereas a hierarchical clustering results in a nested series of partitions. Each of these can either be a hard clustering or a soft one. In a hard clustering, every object may belong to exactly one cluster. In soft clustering, the membership is fuzzy and examples may belong to several clusters with a fractional degree of membership in each. Clustering techniques have proven to be useful in understanding the structure of data, and a variety of document clustering algorithms have been proposed in the literature (see Greene, 2006, for a good review). Deterministic clustering algorithms are those that produce stable clusters which are defined as clusters that "*can be confirmed and reproduced to a* *high degree*" (Mucha, 2006). The word *deterministic* refers to the fact that, as we shall see, consistent subset of centres or representative examples can be identified in many runs of the clustering no matter what the initial clustering and examples order are. Thus deterministic clustering is more stable which can be confirmed and reproduced every time when running it. Deterministic clustering algorithms can be employed to generate the initial training set to make the comparison between different active learning techniques reliable and meaningful.

The remainder of this section will describe the clustering techniques used in our experiments. The non-deterministic algorithms described are *k-Means*, *KMeans+ME*, and *k-Medoids*, all of which have been used in active learning systems before. In the descriptions of these algorithms we will highlight the sources of their non-determinism. The deterministic algorithms described are *furthest-first-traversal* (FFT), *agglomerative hierarchical clustering* (AHC), and *affinity propagation clustering* (APC). To the best of our knowledge, these have not been used in active learning systems for initial training set selection before.

C.1.1 k-Means Clustering

The k-Means algorithm (Duda & Hart, 1973) groups a collection of examples into k clusters so as to minimise the sum of squared distances to the cluster centres. It can be implemented as a simple procedure that initially selects k random *centroids*, assigns each example to the cluster whose centroid is closest, and then calculates a new centroid for each cluster. Examples are reassigned to clusters and new centroids are re-calculated repeatedly until there is no change in clusters. The examples closest to the cluster centroids are selected as the members of the initial training set. This method can be depicted as follows:

Step 1: Set k to the predefined size of initial training set.

Step 2: Randomly pick k seed points as the centroids of the k clusters.

Step 3: Put each remaining example to the cluster with the nearest centroid.

Step 4: Update the centroids as the mean of clusters. Re-calculate the distance from each example to the centroid of each cluster. If the distance between the example and its current centroid is not the smallest, then re-assign it to the cluster with the smallest centroid.

Step 5: Repeat Step 3 and Step 4 until convergence is achieved.

Step 6: Select the example closest to the centroid of each cluster and add it to the initial training set.

The k-Means clustering has the advantage of simplicity and efficiency. However, a major problem with it is its sensitivity to the initial selection of seeds. Moreover, centroids in k-Means are the mean of clusters which are easily influenced by outliers and other extreme values. It has been shown that the performance of k-Means depends on the initial clustering and examples order (na *et al.*, 1999). The non-determinism in k-Means is introduced by the fact that the starting centroids are randomly selected. Different starting centroids can result in vastly different clusterings of the data, and this is exacerbated when the number of clusters k is large or when the data is high-dimensional. Although there have been efforts at making k-Means clustering deterministic (Likas *et al.*, 2001; Su & Dy, 2004), there is no agreed best technique for doing this and so the problem remains.

C.1.2 KMeans+ME approach

KMeans+ME method was proposed by Kang *et al.* (2004) for selecting initial training examples for active learning. Not only the examples closest to the cluster's centroids but also the virtual centroids of the clusters (namely *model examples*) are included in the initial training set. The KMeans+ME method can be depicted as follows:

Step 1: Set k to the predefined size of initial training set.

Step 2: Select k examples randomly and take them as initial seeds to group all the unlabelled examples into k clusters.

Step 3: Once the k-means algorithm converges, the representative example closest to the centroid of each cluster is labelled and put into the initial training set.

Step 4: Since the representative example is closest to the centroid in each cluster, the same label as the closest representative example is applied to the corresponding centroid of each cluster. Those labelled centroids are called model examples.

Step 5: Add the k model examples to the initial training set which result in an initial training set with size of $2 * k$.

It has been shown by Kang *et al.* (2004) that inclusion of the model examples in the initial training set results in even more enhancement of learning performance compared to the method of using k-Means clustering only. Since k-Means clustering is used, it also has the problem of being non-deterministic.

C.1.3 k-Medoids Clustering

The k-Medoids algorithm (Kaufman & Rousseeuw, 1990) is similar to k-Means except that it uses actual examples, *medoids*, as the centre of each cluster instead of artificially generated examples (centroids). The k-Medoids algorithm groups the data into sets of examples by finding k representatives (medoids) $m_1, m_2, \ldots, m_k$ of the dataset and assigning examples to their nearest medoids. The k medoids are found in an iterative way to minimize the sum of the distance from the examples to the nearest medoids. After the k-Medoids algorithm converges the k medoids are used as the initial training examples. The algorithm can be explained as follows. The random selection of the initial k medoids is again the source of non-determinism.

Step 1: Set k to the predefined size of initial training set.

Step 2: Choose k medoids from the dataset randomly.

Step 3: Assign all examples to their closest medoids.

Step 4: Look for a better medoid by swapping one medoid with another example and calculating the distance between it and all examples in the cluster. If the new distance is smaller then set the example to be the medoid and re-partition the examples.

Step 5: Repeat Step 4 until convergence is achieved.

Step 6: Select the medoids and add them to the initial training set.

C.1.4 Furthest-First-Traversal

The *Furthest-First-Traversal* (FFT) clustering technique selects the most diverse examples in a dataset as cluster centres. The algorithm begins by selecting the example closest to the centre of the dataset and then iteratively chooses the example that is located furthest away from the current *centres* as the next centre. When using FFT for the active learning initial training set selection, the first example can be chosen in a deterministic way that makes sense for the dataset (e.g. the document closest to the dataset mean vector, or the longest document). Next is the example furthest from the first centre. Third is the example furthest from both previous centres, and so on until k centres have been identified. In the same way as in the previous approaches the cluster centres found by the FFT algorithm are used as the initial training examples for the active learning process. The algorithm is described as following:

Step 1: Calculate the mean vector of the dataset matrix.

Step 2: Find the example which is closest to the mean vector, namely x_0, that is, its cosine distance from the mean vector is smallest.

Step 3: Find the example which is furthest from x_0, namely x_1.

Step 4: Find the example which is furthest from both x_0 and x_1, namely x_2. And so on until we have k examples.

Step 5: Add the k examples to the initial training set.

Often, *ties* can occur (where more than one example is equi-distant from the current centres) and in these situations the example in the densest area of the dataset is preferred. The density of example x_i is measured by the number of neighbouring examples within a region (specified by a threshold δ, typically set to the mean of the pair-wise distances) that have x_i as its centre. A standardised approach to handling ties ensures that the FFT algorithm remains deterministic.

The FFT algorithm has been used before in active learning (Baram *et al.*, 2004), but as part of a novel selection strategy rather than to prime the initial training set.

C.1.5 Agglomerative Hierarchical Clustering

Agglomerative Hierarchical Clustering (AHC) (Voorhees, 1986) is a bottom-up clustering method which constructs a tree of clusters. Each example is initially assigned to its own individual cluster and the procedure repeatedly combines the two closest clusters until there is only one left. Each step creates a level in a dendrogram tree structure. AHC can be used to select k clusters by pruning the tree so as to retain k leaf nodes in the hierarchy. The examples closest to the centres of these clusters are then selected and labelled to be included in the initial active learning training set. The following steps summarize how the AHC is performed.

Step 1: Assign each example to a separate cluster.

Step 2: Evaluate all pairwise inter-cluster distances and update the distance matrix.

Step 3: Identify the pair of clusters with the shortest distance and merge them to a new cluster.

Step 4: Evaluate all distances from this new cluster to all other clusters, and update the matrix.

Step 5: Repeat Step 3 and Step 4 until only k clusters left.

Step 6: Select the centres of clusters and add them to the initial training set.

A variety of agglomerative hierarchical clustering algorithms have been proposed using different strategies to calculate distance between two clusters. Greene (2006) found *Min-Max linkage* (Ding & He, 2002) to work well on textual data and so this approach is used in our experiments.

C.1.6 Affinity Propagation Clustering

Affinity Propagation Clustering (APC) (Frey & Dueck, 2007) is a clustering algorithm that identifies *exemplars*, or centres, that best represent the dataset and forms clusters of examples around these centres. It operates by simultaneously considering all examples in a collection as potential exemplars, and exchanging messages indicative of the suitability of an example as an exemplar between them until a good set of exemplars emerges. Affinity propagation clustering works on a similarity matrix that is pre-calculated before the start of clustering. These similarities are represented by $s(i, k)$ which indicates how well example k is suited to be the exemplar for example i. Then a priori suitability of example k to serve as an exemplar is namely the *preference* of example k. The preference in affinity propagation clustering is represented by $s(k, k)$ and can be used to control how many examples will be selected as exemplars. The preferences are placed on the diagonal of the similarity matrix S.

There are two kinds of messages exchanged between examples, *responsibility* $r(i, k)$ and *availability* $a(i, k)$. The responsibilities $r(i, k)$ are sent from example i

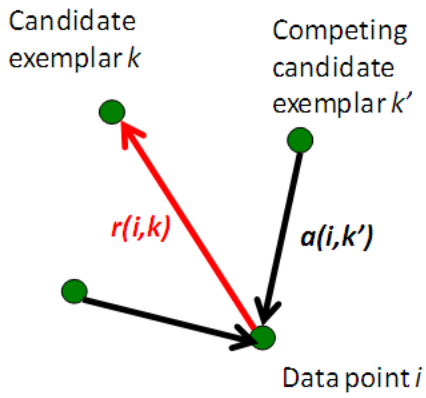


Figure C.1: Sending responsibilities (adopted from Frey & Dueck (2007))

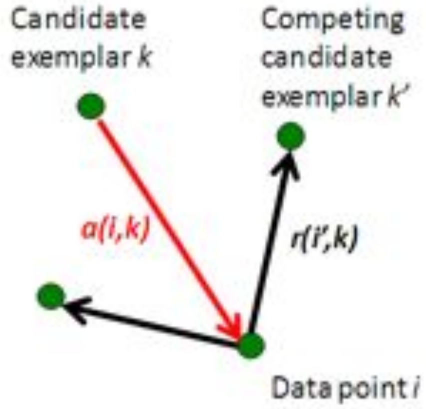


Figure C.2: Sending availabilities (adopted from Frey & Dueck (2007))

to candidate exemplar k and indicate the accumulated evidence for how strongly each example i favors the example k as the exemplar for it, taking into account other candidate exemplar k'. Figure C.1 shows the sending of responsibilities. The availabilities $a(i, k)$ are sent from candidate exemplar k to example i and indicate the accumulated evidence for how appropriate it would be for example i to choose k as its exemplar, taking into account the support from other examples that example k should be an exemplar. Figure C.2 shows the sending of availabilities. Exemplar decisions are monitored by combining availabilities and responsibilities. The affinity propagation algorithm terminates if it reaches a stopping point. The stopping point is defined by two parameters, namely *convits* and *maxits*. It monitors the exemplar decisions that would be made after each iteration and if these don't change over *convits* iterations, the procedure terminates. In any case, if *maxits* iterations are reached, the procedure terminates.

Affinity propagation clustering has been shown to be a deterministic technique that can obtain better solutions than k-medoids, spectral clustering, Gaussian mixture modeling and hierarchical clustering (Frey & Dueck, 2005). But the efficient and stable clustering technique has not, to our knowledge, been applied in active learning. In the following, we employ affinity propagation clustering as an initial training set construction method, i.e. we consider the exemplars from the affinity propagation clustering as initial training examples to seed the active learner. The method can be described as follows:

Step 1: Calculate similarity $s(i, k)$ between all pairs of example i and k in the dataset and build the similarity matrix S.

Step 2: Set the preferences all equal to p.

Step 3: Simultaneously consider all the data points as potential exemplars (cluster centres) by viewing each data point as a node in a network.

Step 4: Two kinds of messages: responsibility $r(i, k)$ and availability $a(i, k)$ are calculated.

Step 5: Recursively exchange messages between examples by updating $r(i, k)$ and $a(i, k)$ as in Equation C.1 and Equation C.2. More details can be found in

Frey & Dueck (2007).

$$r(i,k) \leftarrow s(i,k) - \max_{k' \ s.t. \ k' \neq k} \{a(i,k') + s(i,k')\} \tag{C.1}$$

$$a(i,k) \leftarrow \min \left\{ 0, r(k,k) + \sum_{i' \ s.t. \ i' \notin \{i,k\}} \max\{0, r(i',k)\} \right\} \tag{C.2}$$

Step 6: Combine $r(i,k)$ and $a(i,k)$ to find the exemplar when $a(i,k) + r(i,k)$ is maximized.

Step 7: Repeat Step 4 to Step 6 until convergence is achieved.

Step 8: Add the obtained exemplars to the initial training set.

The affinity propagation clustering algorithm has a parameter preference, p, which is used to control the number of clusters obtained. Broadly, a higher value of p results in more clusters and a lower value of p in less. To find a specific number of clusters in a dataset the value of p must be tuned experimentally, which is a disadvantage of the technique. In order to get a predefined size of initial training set, we can start with the value of $median_{i,k:i \neq k} s(i,k)$ as suggested in Frey & Dueck (2007) and to see how many exemplars we get and adjust the preference until we get the predefined number of exemplars. Table C.1 lists the preference values for 10 exemplars in four datasets used in our experiments.

Table C.1: Preference values.

Dataset	**Preference**
WinXwin	-1.55
Comp	-2
Reuters	-1.8
RCV1	-4

References

Abe, N. & Kudo, M. (2006). Non-parametric classifier-independent feature selection. *Pattern Recognition*, **39**, 737–746. 15

Abe, N. & Mamitsuka, H. (1998). Query learning strategies using boosting and bagging. In *Proceedings of the Fifteenth International Conference on Machine Learning*, 1–9, Morgan Kaufmann Publishers Inc. 71

Abe, N., Zadrozny, B. & Langford, J. (2006). Outlier detection by active learning. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 504–509. 33

Abramson, Y. & Freund, Y. (2005). Semi-automatic visual learning (SEVILLE): Tutorial on active learning for visual object recognition. In *Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*. 34, 62

Airoldi, E.M., Cohen, W.W. & Fienberg, S.E. (2004). Bayesian models for frequent terms in text. Tech. rep., Carnegie Mellon University. 25

Allwein, E.L., Schapire, R.E. & Singer, Y. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers. In *Proceedings of the 17th International Conference on Machine Learning*, 9–16. 23

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. 58

Angluin, D. (1988). Queries and concept learning. *Machine Learning*, **2**, 319–342. 32, 35

Anupam, V., Bajaj, C., Schikore, D. & Schikore, M. (1994). Distributed and collaborative visualization. *Computer*, **27**, 37–43. 60

Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R. & Wu, A.Y. (1998). An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, **45**, 891–923. 21

Ayache, S. & Quénot, G. (2007a). Evaluation of active learning strategies for video indexing. *Image Communication*, **22**, 692–704. 73

Ayache, S. & Quénot, G. (2007b). TRECVID 2007 collaborative annotation using active learning. In *Proceedings of the TRECVID 2007 Workshop*. 73

Baeza-Yates, R. & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley, 1st edn. 18, 20, 54

Baldridge, J. & Osborne, M. (2003). Active learning for HPSG parse selection. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, 17–24, Association for Computational Linguistics, Edmonton, Canada. 33

Baldridge, J. & Osborne, M. (2004). Active learning and the total cost of annotation. In *Proceedings of EMNLP 2004*, 9–16. 65, 67, 72

Baram, Y., El-Yaniv, R. & Luz, K. (2004). Online choice of active learning algorithms. *Journal of Machine Learning Research*, **5**, 255–291. 36, 49, 54, 55, 59, 71, 203

Baum, E. & Lang, K. (1992). Query learning can work poorly when a human oracle is used. In *Proceedings of the IEEE International Joint Conference on Neural Networks*. 35

Becker, M. & Osborne, M. (2005). A two-stage method for active learning of statistical grammars. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, 991–996. 42

Blum, A. & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, 92–100. 70

Bottou, L., Cortes, C., Denker, J., Drucker, H., Guyon, I., Jackel, L., LeCun, Y., Muller, U., Sackinger, E., Simard, P. & Vapnik, V. (1994). Comparison of classifier methods: a case study in handwritten digit recognition. In

Proceedings of the 12th International Conference on Pattern Recognition, 77–87. 23

Bowring, J.F., Rehg, J.M. & Harrold, M.J. (2004). Active learning for automatic classification of software behavior. In *Proceedings of the 2004 ACM SIGSOFT international symposium on Software testing and analysis, ISSTA '04*, vol. 29, 195–205, Boston, Massachusetts, USA. 34

Brinker, K. (2003). Incorporating diversity in active learning with support vector machines. In *Proceedings of the Twentieth International Conference on Machine Learning*, 59–66. 55

Brinker, K. (2006). On active learning in multi-label classification. In M. Spiliopoulou, R. Kruse, C. Borgelt, A. Nürnberger & W. Gaul, eds., *From Data and Information Analysis to Knowledge Engineering*, 206–213, Springer-Verlag, Berlin/Heidelberg. 73

Burges, C.J.C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, **2**, 121–167. 22

Campbell, C., Cristianini, N. & Smola, A.J. (2000). Query learning with large margin classifiers. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 111–118, Morgan Kaufmann Publishers Inc. 57

Cebron, N. & Berthold, M. (2008). Active learning for object classification: from exploration to exploitation. *Data Mining and Knowledge Discovery*. 40, 46, 56, 57, 109

Cebron, N. & Berthold, M.R. (2005). Mining of cell assay images using active semi-supervised clustering. In *Proceedings of the ICDM 2005 Workshop on Computational Intelligence in Data Mining*, 63–69. 33

Cebron, N. & Berthold, M.R. (2006). Adaptive active classification of cell assay images. *European Conference on Principles and Practice of Knowledge Discovery (PKDD), LNCS*, **4213**, 79–90. 33, 51, 52

Chang, J.S., Luo, Y.F. & Su, K.Y. (1992). GPSM: A generalized probabilistic semantic model for ambiguity resolution. In *Proceedings of the 30th annual meeting on Association for Computational Linguistics, ACL '92*, 177–184. 29

Cheetham, W. (2000). Case-based reasoning with confidence. In E. Blanzieri & L. Portinale, eds., *5th European Workshop on Case-Based Reasoning*, vol. 1898 of *LNCS*, 15–25, Springer. 48

Cheetham, W. & Price, J. (2004). Measures of solution accuracy in case-based reasoning systems. In P. Funk & P. González-Calero, eds., *7th European Conference on Case-Based Reasoning (ECCBR 2004)*, vol. 3155 of *LNAI*, 106–118, Springer. 49

Chen, K. & Liu, L. (2006). iVIBRATE: Interactive visualization-based framework for clustering large datasets. *ACM Transactions on Information Systems (TOIS)*, **24**, 245–294. 61

Chernoff, H. (1973). The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, **68**, 361–368. 60

COHN, D.A. (1996). Neural network exploration using optimal experiment design. *Neural Networks*, **9**, 1071–1083. 35

CONNOR, M. & KUMAR, P. (2010). Fast construction of k-Nearest neighbor graphs for point clouds. *IEEE Transactions on Visualization and Computer Graphics*, **16**, 599–608. 21

COOPER, S., HERTZMANN, A. & POPOVIĆ, Z. (2007). Active learning for real-time motion controllers. *ACM Transactions on Graphics (TOG)*, **26**, 5–11. 34

CORD, M., GOSSELIN, P.H. & PHILIPP-FOLIGUET, S. (2007). Stochastic exploration and active learning for image retrieval. *Image and Vision Computing*, **25**, 14–23. 34

CORTES, C. & VAPNIK, V. (1995). Support-Vector networks. *Machine Learning*, **20**, 273–297. 18

COVER, T. & HART, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, **13**, 21–27. 18, 19

CULOTTA, A. & MCCALLUM, A. (2005). Reducing labeling effort for structured prediction tasks. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 2*, 746–751, AAAI Press, Pittsburgh, Pennsylvania. 72

CUNNINGHAM, P. & DELANY, S.J. (2007). k-nearest neighbour classifiers. Tech. rep., University College Dublin and Dublin Institute of Technology. 21

Dagan, I. & Engelson, S.P. (1995). Committee-based sampling for training probabilistic classifiers. In *Proceedings of 1995 International Conference on Machine Learning*, 150–157. 42

Dagli, C., Rajaram, S. & Huang, T. (2005). Combining diversity-based active learning with discriminant analysis in image retrieval. In *Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05)*, vol. 2, 173–178. 32, 34

Dagli, C., Rajaram, S. & Huang, T. (2006). Leveraging active learning for relevance feedback using an information theoretic diversity measure. *Lecture Notes in Computer Science*, **4071**, 123–132. 34

Dasarathy, B.V. (1995). Nearest unlike neighbor (nun): an aid to decision confidence estimation. *Optical Engineering*, **34**, 2785–2792. 48

Dasgupta, S. & Hsu, D. (2008). Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*. 49, 51

Dasgupta, S. & Ng, V. (2009). Mine the easy, classify the hard: A semi-supervised approach to automatic sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language*, vol. 2, 701–709, Suntec, Singapore. 39, 45

Davy, M. & Luz, S. (2007a). Active learning with history-based query selection for text categorisation. *Lecture Notes in Computer Science*, **4425**, 695–698. 41

Davy, M. & Luz, S. (2007b). Dimensionality reduction for active learning with nearest neighbour classifier in text categorisation problems. In *Proceedings of the Sixth International Conference on Machine Learning and Applications*, 292–297, IEEE Computer Society. 15, 46

de Oliveira, M.C.F. & Levkowitz, H. (2003). From visual data exploration to visual data mining: A survey. *IEEE Transactions on Visualization and Computer Graphics*, **9**, 378–394. 60

Delany, S.J., Cunningham, P. & Doyle, D. (2005a). Generating estimates of classification confidence for a case-based spam filter. In *Proceedings of the 6th International Conference on Case-Based Reasoning (ICCBR 2005)*, vol. 3620 of *LNAI*, 170–190, Springer. 49, 89, 90, 91, 93, 95

Delany, S.J., Cunningham, P., Tsymbal, A. & Coyle, L. (2005b). A case-based technique for tracking concept drift in spam filtering. *Knowledge-Based Systems*, **18**, 3–16. 82, 150

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, **7**, 1–30. 16, 29

Devijver, P.A. & Kittler, J. (1982). *Pattern Recognition: A Statistical Approach*. Prentice/HallInternational. 29

Dietterich, T.G. (1998). Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, **10**, 1895–1923. 29

Dimitrakakis, C., Krohn, C., Hartmann, S., Isberner, G., Hartmann, S. & Isberner, G. (2008). Cost-minimising strategies for data labelling: Optimal stopping and active learning. In *Lecture Notes in Computer Science*, vol. 4932, 96–111, Springer. 72

Ding, C. & He, X. (2002). Cluster merging and splitting in hierarchical clustering algorithms. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, 139, IEEE Computer Society. 204

Domingos, P. & Pazzani, M. (1997). On the optimality of the simple bayesian classifier under Zero-One loss. *Machine Learning*, **29**, 103–130. 24

Dong, A. & Bhanu, B. (2003). Active concept learning for image retrieval in dynamic databases. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, 90, IEEE Computer Society. 57

Donmez, P. & Carbonell, J.G. (2008a). Paired sampling in density-sensitive active learning. In *Proceedings of the 10th International Symposium on Artificial Intelligence and Mathematics, ISAIM '08*. 56

Donmez, P. & Carbonell, J.G. (2008b). Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08*, 619–628, Napa Valley, California, USA. 72

Donmez, P., Carbonell, J. & Bennett, P. (2007). Dual strategy active learning. In *Proceedings of the 18th European conference on Machine Learning*, 116–127. 50, 52, 71

Donmez, P., Carbonell, J.G. & Schneider, J. (2009). Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 259–268, ACM, Paris, France. 57

Dredze, M. & Crammer, K. (2008). Active learning with confidence. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, 233–236, Association for Computational Linguistics. 47

Drucker, H., Wu, D. & Vapnik, V. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, **10**, 1048–1054. 15

Duda, R.O. & Hart, P.E. (1973). *Pattern Classification and Scene Analysis*. John Wiley & Sons Inc. 198

Duda, R.O., Hart, P.E. & Stork, D.G. (2000). *Pattern Classification*. Wiley-Interscience, 2nd edn. 18, 19

Dumais, S., Platt, J., Heckerman, D. & Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on Information and knowledge management*, 148–155, ACM, Bethesda, Maryland, United States. 15

Eades, P. (1984). A heuristic for graph drawing. *Congressus Nutnerantiunt*, **42**, 149–160. 63

Ertekin, S. (2005). Can computers learn faster. In *Proceedings of the Second COE Research Symposium*, The Pennsylvania State University, University Park. 45

Ertekin, S., Huang, J. & Giles, C.L. (2007). Active learning for class imbalance problem. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '07*, 823–824, Amsterdam, Netherlands. 58

Esuli, A. & Sebastiani, F. (2009). Active learning strategies for Multi-Label text classification. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval, ECIR '09*, 102–113. 73

Eyheramendy, S., Lewis, D.D. & Madigan, D. (2003). On the Naïve Bayes model for text categorization. In *Proceedings of the Ninth InternationalWorkshop on Artificial Intelligence and Statistics.*, 332—339. 25

Fellbaum, C., ed. (1998). *WordNet: An Electronic Lexical Database*. MIT Press. 176

Freund, Y., Seung, H., Shamir, E. & Tishby, N. (1997). Selective sampling using the query by committee. *Machine Learning*, **28**, 133–168. 35

Frey, B.J. & Dueck, D. (2005). Mixture modeling by affinity propagation. *Advances in Neural Information Processing Systems*, **2**, 379–386. 206

Frey, B.J. & Dueck, D. (2007). Clustering by passing messages between data points. *Science*, **315**, 972–976. 181, 204, 205, 207

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, **11**, 86–92. 29

Friedman, N., Geiger, D. & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, **29(2-3)**, 131–163. 24

Fujii, A., Tokunaga, T., Inui, K. & Tanaka, H. (1998). Selective sampling for example-based word sense disambiguation. *Computational Linguistics*, **24**, 573–597. 41, 53

Garcia, V., Debreuve, E. & Barlaud, M. (2008). Fast k nearest neighbor search using GPU. In *Proceedings of the Computer Vision and Pattern Recognition Workshops*, 1–6. 21

Gey, F.C. (1994). Inferring probability of relevance using the method of logistic regression. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '94*, 222—231. 19

Ghani, R. (2001). Combining labeled and unlabeled data for text classification with a large number of categories. In N. Cercone, T.Y. Lin & X. Wu, eds., *Proceedings of the IEEE International Conference on Data Mining*, 597–598, IEEE Computer Society, San Jose, USA. 44

Ghayoomi, M. (2010). Using variance as a stopping criterion for active learning of frame assignment. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*. 33

Godbole, S., Harpale, A., Sarawagi, S. & Chakrabarti, S. (2004). HIClass: Hyper-interactive text classification by interactive supervision of document and term labels. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 546–548. 45

Gosselin, P.H. & Cord, M. (2005). Active learning techniques for user interactive systems: Application to image retrieval. In *Proceedings of the International Workshop on Machine Learning techniques for processing MultiMedia content*. 32

Greene, D. (2006). *A State-of-the-Art Toolkit for Document Clustering*. Ph.D. thesis, Trinity College Dublin. 96, 134, 145, 197, 204

Greiff, W.R. & Ponte, J.M. (2000). The maximum entropy approach and probabilistic IR models. *ACM Transactions on Information Systems (TOIS)*, **18**, 246–287. 19

Guyon, I., Cawley, G., Dror, G. & Lemaire, V. (2010). Design and analysis of the WCCI 2010 active learning challenge. In *Proceedings of the IEEE/INNS International Joint Conference on Neural Networks (IJCNN-2010)*, 1–8. 174, 180, 182

HAERTEL, R.A., SEPPI, K.D., RINGGER, E.K. & CARROLL, J.L. (2008). Return on investment for active learning. In *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*. 72

HANLEY, J.A. & MCNEIL, B.J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, **143**, 29–36. 28

HASENJAGER, M. & RITTER, H. (1998). Active learning with local models. *Neural Processing Letters*, **7**, 107–117. 46

HE, J. & CARBONELL, J. (2007). Nearest-neighbor-based active learning for rare category detection. In J.C. Platt, D. Koller, Y. Singer & S.T. Roweis, eds., *Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems*, vol. 20, MIT Press, Vancouver, British Columbia, Canada. 33, 107

HECHENBICHLER, K. & SCHLIEP, K. (2004). Weighted k-nearest-neighbor techniques and ordinal classification. *SFB Discussion Paper 399*. 21

HO, S.S. & WECHSLER, H. (2003). Transductive confidence machine for active learning. In *Proceedings of the International Joint Conference on Neural Networks*, vol. 2, 1435–1440. 48

HOI, S.C., JIN, R. & LYU, M.R. (2009a). Batch mode active learning with applications to text categorization and image retrieval. *IEEE Transactions on Knowledge and Data Engineering*, **21**, 1233–1248. 59

Hoi, S.C.H., Jin, R. & Lyu, M.R. (2006a). Large-scale text categorization by batch mode active learning. In *Proceedings of the 15th international conference on World Wide Web*, 633–642, ACM, Edinburgh, Scotland. 37, 43

Hoi, S.C.H., Jin, R., Zhu, J. & Lyu, M.R. (2006b). Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd international conference on Machine learning*, 417–424, ACM, Pittsburgh, Pennsylvania. 33

Hoi, S.C.H., Jin, R., Zhu, J. & Lyu, M.R. (2009b). Semisupervised SVM batch mode active learning with applications to image retrieval. *ACM Transactions on Information Systems (TOIS)*, **27**, 1–29. 32

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, **24**, 498–520. 60

H.S.Seung, M.Opper & H.Sompolinsky (1992). Query by committee. In *Proceedings of the Fifth Workshop on Computational Learning Theory*, 287–294, Morgan Kaufmann, San Mateo, CA. 42

Hsu, C.W. & Lin, C.J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, **13**, 415–425. 23

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)*. 10, 65, 78, 88, 152, 174

Hu, R., Delany, S.J. & Mac Namee, B. (2009). Sampling with confidence: Using k-NN confidence measures in active learning. In *Proceedings of the UKDS Workshop at 8th International Conference on Case-based Reasoning (ICCBR 09)*, 181–192. 6, 9, 11, 89, 142

Hu, R., Delany, S.J. & Mac Namee, B. (2010a). EGAL: exploration guided active learning for TCBR. In *Proceedings of the 18th International Conference on Case-Based Reasoning, ICCBR 2010*, vol. 6176 of *Lecture Notes in Computer Science*, 156–170, Springer Berlin / Heidelberg, Alessandria, Italy. 6, 9, 11, 50, 63

Hu, R., Lindstrom, P., Delany, S.J. & Mac Namee, B. (2010b). Exploring the frontier of uncertainty space. In *AISTATS 2010 Workshop on Active Learning and Experimental Design*. 6, 9, 11

Hu, R., Mac Namee, B. & Delany, S.J. (2010c). 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)*, 26–31, AAAI. 6, 10, 11, 96, 102, 114, 151

Hua, X.S. & Qi, G.J. (2008). Online multi-label active learning for large-scale multimedia annotation. Tech. Rep. MSR-TR-2008-103, Microsoft Research. 73

Hwa, R. (2004). Sample selection for statistical parsing. *Computational Linguistics*, **30**, 253–276. 40

Imamura, M., Takayama, Y., Kaji, N., Toyoda, M. & Kitsuregawa, M. (2009). A combination of active learning and semi-supervised learning starting with positive and unlabeled examples for word sense disambiguation: an empirical study on Japanese web search query. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, 61–64, Association for Computational Linguistics, Suntec, Singapore. 33, 47

Jiang, J.J. & Conrath, D.W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference Research on Computational Linguistics (ROCLING X)*, 19–33, Taiwan. 176

Joachims, T. (1997). Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning, ECML '98*, vol. 1398, 137–142, Springer. 18, 23

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In I. Bratko & S. Dzeroski, eds., *Proceedings of the 16th International Conference on Machine Learning*, 200–209, Morgan Kaufmann Publishers, San Francisco, US, Bled, SL. 15, 70

Jones, R., Ghani, R., Mitchell, T. & Riloff, E. (2003). Active learning for information extraction with multiple view feature sets. In *Proceedings of the ECML-2004 Workshop on Adaptive Text Extraction and Mining (ATEM-2003)*. 34

Juszczak, P. & Duin, R. (2003). Selective sampling methods in One-Class classification problems. In *Proceedings of the 2003 joint international conference on*

Artificial neural networks and neural information processing, 140–148, Springer-Verlag, Berlin, Heidelberg. 33

Kang, J., Ryu, K. & Kwon, H. (2004). Using cluster-based sampling to select initial training set for active learning in text classification. *Advances in Knowledge Discovery and Data Mining*, **3056**, 384–388. 39, 96, 135, 139, 200

Kapoor, A., Grauman, K., Urtasun, R. & Darrell, T. (2007a). Active learning with gaussian processes for object categorization. In *IEEE 11th International Conference on Computer Vision (ICCV 2007)*, 1–8. 34

Kapoor, A., Horvitz, E. & Basu, S. (2007b). Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *International Joint Conference on Artificial Intelligence*. 34, 72

Kaufman, L. & Rousseeuw, P.J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience. 201

Keim, D. (2000). Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, **6**, 59–78. 60

Keim, D.A., Bak, P., Bertini, E., Oelke, D., Spretke, D. & Ziegler, H. (2010). Advanced visual analytics interfaces. In G. Santucci, ed., *Proceedings of the International Conference on Advanced Visual Interfaces*, AVI '10, 3–10, ACM, New York, NY, USA. 60

KIBLER, D. & AHA, D. (1987). Learning representative exemplars of concepts: An initial case study. In *Proceedings of the Fourth International Workshop on Machine Learning*, 24–30, Morgan Kaufmann, Irvine, CA. 21

KOHAVI, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th international joint conference on Artificial intelligence*, vol. 2 of *IJCAI '95*, 1137–1145. 29

KOHONEN, T. (2000). *Self-Organizing Maps*. Springer, 3rd edn. 60

KOLLER, D. & SAHAMI, M. (1997). Hierarchically classifying documents using very few words. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 170–178, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 25

KONTKANEN, P., LAHTINEN, J., MYLLYMÄKI, P., SILANDER, T. & TIRRI, H. (2000). Supervised model-based visualization of high-dimensional data. *Intelligent Data Analysis*, **4**, 213–227. 60

KRISHNAKUMAR, A. (2007). Active learning literature survey. Tech. rep., University of California, Santa Cruz. 42

KRITHARA, A., GOUTTE, C., MASSIH, A. & RENDERS, J.M. (2006). Reducing the annotation burden in text classification. In *Proceedings of the First International Conference on Multidisciplinary Information Sciences and Technologies (InSciT 2006)*, Mérida, Spain. 70

Langley, P., , Iba & Thompson, K. (1992). An analysis of bayesian classifiers. In *Proceedings of the tenth international conference on Artificial intelligence*, 223–228, AAAI Press. 24

Lanquillon, C. (2000). Partially supervised text classification: Combining labeled and unlabeled documents using an EM-like scheme. In *Proceedings of the 11th European Conference on Machine Learning*, 229–237. 70

Larkey, L.S. & Croft, W.B. (1996). Combining classifiers in text categorization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, 289–297, ACM, New York, NY, USA. 25

Laws, F. & Schütze, H. (2008). Stopping criteria for active learning of named entity recognition. In *Proceedings of the 22nd International Conference on Computational Linguistics*, vol. 1, 465–472. 33, 58

Lecerf, L. & Chidlovskii, B. (2009). Visalix: A web application for visual data analysis and clustering. In *Demonstrations Track of the 15th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, 28–31, Paris, France. 62

Lee, M.S., Rhee, J.K., Kim, B.H. & Zhang, B.T. (2009). AESNB: Active example selection with naive bayes classifier for learning from imbalanced biomedical data. In *Proceedings of the 2009 Ninth IEEE International Conference on Bioinformatics and Bioengineering*, 15–21. 38

Lewis, D. & Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, 148–156, Morgan Kaufmann. 65, 147

Lewis, D.D. (1991). Evaluating text categorization. In *Proceedings of the workshop on Speech and Natural Language, HLT '91*, 312–318, Association for Computational Linguistics, Morristown, NJ, USA. 25

Lewis, D.D. (1992). *Representation and Learning in Information Retrieval*. Ph.D. thesis, Department of Computer and Information Science, University of Massachusetts. 27

Lewis, D.D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. *Lecture Notes in Computer Science*, **1398**, 4–15. 24

Lewis, D.D. & Gale, W.A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual International ACM SIGIR conference on Research and Development in Information Retrieval*, 3–12, Springer-Verlag NY. 36, 40

Lewis, D.D. & Ringuette, M. (1994). A comparison of two learning algorithms for text categorization. In *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval, SDAIR '94*, 81–93, Las Vegas, USA. 24

Lewis, D.D., Yang, Y., Rose, T.G. & Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, **5**, 361–397. 82, 150

Li, B., Lu, Q. & Yu, S. (2004a). An adaptive k-nearest neighbor text categorization strategy. *ACM Transactions on Asian Language Information Processing (TALIP)*, **3**, 215–226. 49

Li, M. (2005). *Confidence-Based Classifier Design and Its Applications*. Ph.D. thesis, Oakland University. 47

Li, M. & Sethi, I.K. (2004). SVM-based classifier design with controlled confidence. In *Proceedings of the17th International Conference on Pattern Recognition, ICPR '04*, vol. 1, 164–167, IEEE Computer Society. 47

Li, M. & Sethi, I.K. (2006a). Confidence-based active learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**, 1251–1261. 41, 47

Li, M. & Sethi, I.K. (2006b). Confidence-based classifier design. *Pattern Recognition*, **39**, 1230–1240. 47

Li, X., Wang, L. & Sung, E. (2004b). Multilabel SVM active learning for image classification. In *Proceedings of 2004 International Conference on Image Processing, ICIP '04*, vol. 4, 2207–2210. 34, 73

Li, Y. & Guo, L. (2007). An active learning based TCM-KNN algorithm for supervised network intrusion detection. *Computers and Security*, **26**, 459–467. 33, 46, 48

Liere, R. & Tadepalli, P. (1997). Active learning with committees for text categorization. In *Proceedings of the fourteenth national conference on artificial* *intelligence and ninth conference on Innovative applications of artificial intelligence*, 591–596. 42

Likas, A., Vlassis, N. & Verbeek, J.J. (2001). The global K-Means clustering algorithm. *Pattern Recognition*, **36**, 451–461. 199

Lin, H. & Bilmes, J. (2009). How to select a good training-data subset for transcription submodular active selection for sequences. Tech. rep., University of Washington. 34

Lindenbaum, M., Markovitch, S. & Rusakov, D. (2004). Selective sampling for nearest neighbor classifiers. *Machine Learning*, **54**, 125–152. 41, 46

Lindstrom, P., Hu, R., Delany, S.J. & Mac Namee, B. (2010). SVM based active learning with exploration. In *AISTATS 2010 Workshop on Active Learning and Experimental Design*. 10

Liu, Y. (2004). Active learning with support vector machine applied to gene expression data for cancer classification. *Journal of Chemical Information and Computer Sciences*, **44**, 1936–1941. 33

Loureiro, O. & Siegelmann, H. (2005). Introducing an active cluster-based information retrieval paradigm. *Journal of the American Society for Information Science and Technology*, **56**, 1024–1030. 34

Luo, T., Kramer, K., Goldgof, D.B., Hall, L.O., Samson, S., Remsen, A. & Hopkins, T. (2005). Active learning to recognize multiple types of plankton. *Journal of Machine Learning Research*, **6**, 589–613. 33, 73

Ma, A., Patel, N., Li, M. & Sethi, I. (2006). Confidence based active learning for whole object image segmentation. In *Proceedings of 2006 International Workshop on Multimedia Content Representation, Classification and Security, MRCS '06*, vol. 4105 of *Lecture Notes in Computer Science*, 753–760, Springer Berlin / Heidelberg, Istanbul, Turkey. 34, 47

Mac Namee, B. & Delany, S.J. (2010). CBTV: Visualising case bases for similarity measure design and selection. In *Proceedings of the 2010 International Conference on Case-based Reasoning, ICCBR '10*, 213–227. 63

Mac Namee, B., Hu, R. & Delany, S.J. (2010). Inside the selection box: Visualising active learning selection strategies. In *NIPS 2010 Workshop on Challenges of Data Visualization*, Whistler, BC. 6, 10, 11, 63, 121

Mandel, M., Poliner, G. & Ellis, D. (2006). Support vector machine active learning for music retrieval. *Multimedia Systems*, **12**, 3–13. 34

Manning, C. & Schutze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts. 29

Martín-Valdivia, M.T., García-Vega, M. & Ureña-López, L.A. (2003). LVQ for text categorization using multilingual linguistic resources. *Neurocomputing*, **55**, 665–679. 18

Massie, S., Craw, S. & Wiratunga, N. (2004). A visualisation tool to explain case-base reasoning solutions for tablet formulation. In *Proceedings of the 24th SGAI International Conference on Innovative Techniques and Applications* *of Artificial Intelligence (AI-2004)*, LNCS, 222–234, Springer, Cambridge, UK. 61

McCallum, A. & Nigam, K. (1998a). A comparison of event models for naive bayes text classification. *Dimension Contemporary German Arts And Letters*, **752**, 41–48. 18, 24, 25

McCallum, A.K. & Nigam, K. (1998b). Employing EM and pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, Morgan Kaufmann. 32, 33, 43, 53

McCarthy, J., Minsky, M.L., Rochester, N. & Shannon, C.E. (1955). A proposal for the dartmouth summer research project on artificial intelligence. Tech. rep., Dartmouth College. 1

Mckenna, E. & Smyth, B. (2001). An interactive visualisation tool for case-based reasoners. *Applied Intelligence*, **14**, 95–114. 61

Melluish, T., Saunders, C., Nouretdinov, I. & Vovk, V. (2001). Comparing the bayes and typicalness frameworks. In *Proceedings of the 12th European Conference on Machine Learning*, vol. 2167, 360–371. 47

Melville, P. & Mooney, R.J. (2004). Diverse ensembles for active learning. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, 74–81, ACM, New York, NY, USA. 54, 71

Mitchell, T. (1997). *Machine Learning*. McGraw Hill Higher Education, 1st edn. 14, 18, 25

Monteleoni, C. & Kaariainen, M. (2007). Online active learning in practice. Tech. rep., MIT Computer Science and Artificial Intelligence Laboratory. 34

Morales, A., Chinellato, E., Fagg, A.H. & del Pobil, A.P. (2004). Active learning for robot manipulation. In *ECAI 2004: Proceedings of the 16th European Conference on Artificial Intelligence*. 33

Morelos-Zaragoza, R.H. (2006). *The art of error correcting coding*. John Wiley and Sons. 23

Moskovitch, R., Nissim, N., Stopel, D., Feher, C., Englert, R. & Elovici, Y. (2007). Improving the detection of unknown computer worms activity using active learning. In *Proceedings of the 30th annual German conference on Advances in Artificial Intelligence*, 489–493, Springer-Verlag, Berlin, Heidelberg. 33

Moskovitch, R., Nissim, N. & Elovici, Y. (2009). Malicious code detection using active learning. In F. Bonchi, E. Ferrari, W. Jiang & B. Malin, eds., *Privacy, Security, and Trust in KDD*, 74–91, Springer-Verlag, Berlin, Heidelberg. 44

Mucha, H. (2006). Finding meaningful and stable clusters using local cluster analysis. In *Data Science and Classification*, Studies in Classification, Data Analysis, and Knowledge Organization, 101–108, Springer Berlin Heidelberg. 198

Muslea, I., Minton, S. & Knoblock, C.A. (2000). Selective sampling with redundant views. In *Proceedings of the Seventeenth National Conference on Arti-*

ficial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, 621–626, AAAI Press. 70

Muslea, I., Minton, S. & Knoblock, C.A. (2002). Active + semi-supervised learning = robust multi-view learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, 435–442, Morgan Kaufmann Publishers Inc. 70

na, J.M.P., Lozano, J.A. & naga, P.L. (1999). An empirical comparison of four initialization methods for the K-Means algorithm. *Pattern Recognition Letters*, **20**, 1027–1040. 199

Nguyen, H.T. & Smeulders, A. (2004). Active learning using pre-clustering. *In Proceedings of the 21st International Conference on Machine Learning*, 623–630. 39, 52, 118

Nguyen, T.T., Binh, N.D. & Bischof, H. (2009). Efficient boosting-based active learning for specific object detection problems. *International Journal of Electrical, Computer, and Systems Engineering*, **3**. 34

Nigam, K. & Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management*, CIKM '00, 86–93, ACM, New York, NY, USA. 44, 70

NIGAM, K. & MCCALLUM, A. (1998). Pool-based active learning for text classification. In *Workshop on Learning from Text and the Web, Conference on Automated Learning and Discovery*. 33, 36

NIGAM, K., MCCALLUM, A., THRUN, S. & MITCHELL, T. (1998). Learning to classify text from labeled and unlabeled documents. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 792–799, AAAI. 44

NIGAM, K., LAFFERTY, J. & MCCALLUM, A. (1999). Using maximum entropy for text classification. In *Proceedings of IJCAI-99 Workshop on Machine Learning for Information Filtering*, 61–67. 15

NIGAM, K., MCCALLUM, A.K., THRUN, S. & MITCHELL, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, **39**, 103–134. 44, 70

NOVAK, B., MLADENIČ, D. & GROBELNIK, M. (2006). Text classification with active learning. In *From Data and Information Analysis to Knowledge Engineering*, 398–405, Springer-Verlag, Berlin/Heidelberg. 32, 38, 44, 57

OSUGI, T., KUN, D. & SCOTT, S. (2005). Balancing exploration and exploitation: A new algorithm for active machine learning. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, 330–337, IEEE Computer Society. 43, 54, 55

PANG, B., LEE, L. & VAITHYANATHAN, S. (2002). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, 79–86. 15

PELLEG, D. & MOORE, A. (2004). Active learning for anomaly and rare-category detection. *In Advances in Neural Information Processing Systems 18*, **16**, 1073—1080. 33

PEREIRA, F. (1993). Distributional clustering of english words. *In Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, 183–190. 41

PLATT, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 61–74. 47

PORTER, M. (1980). An algorithm for suffix stripping. *Program*, **14**, 130–137. 16

PROBST & GHANI (2007). Towards 'interactive' active learning in multi-view feature sets for information extraction. In *Proceedings of the 18th European conference on Machine Learning*, 683–690. 34, 44, 59

PROEDROU, K., NOURETDINOV, I., VOVK, V. & GAMMERMAN, A. (2002). Transductive confidence machines for pattern recognition. In *Proceedings of the 13th European Conference on Machine Learning*, 221–231. 48

PROVOST, F.J., FAWCETT, T. & KOHAVI, R. (1998). The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth In-*

ternational Conference on Machine Learning, 445–453, Morgan Kaufmann Publishers Inc. 28

QI, G.J., SONG, Y., HUA, X.S., ZHANG, H.J. & DAI, L.R. (2006). Video annotation by active learning and cluster tuning. In *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, 114, IEEE Computer Society. 34, 39

QUINLAN, J.R. (1992). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1st edn. 18

RAGHAVAN, H., MADANI, O. & JONES, R. (2006). Active learning with feedback on both features and instances. *Journal of Machine Learning Research*, **7**, 1655–1686. 40, 43, 45, 59

RAGHAVAN, V.V. & WONG, S.K.M. (1986). A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Science*, **37**, 279–87. 16

REICHART, R., TOMANEK, K., HAHN, U. & RAPPOPORT, A. (2008). Multi-Task active learning for linguistic annotations. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, 861–869, Association for Computational Linguistics, Columbus, Ohio, USA. 71

REN, P. & WATSON, B. (2005). Histographs: Interactive visualization of complex data with graphs. Tech. rep., Northwestern University. 60

RENNIE, J. & RIFKIN, R. (2001). Improving multiclass text classification with the support vector machine. Tech. rep., Massachusetts Insititute of Technology. 43

RENNIE, J., SHIH, L., TEEVAN, J. & KARGER, D. (2003). Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the Twentieth International Conference on Machine Learning*. 25

ROY, N. & MCCALLUM, A. (2001). Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the 18th International Conference on Machine Learning*, 441–448, Morgan Kaufmann, San Francisco, CA. 42, 43, 55, 58, 105

ROY, V. & MADHVANATH, S. (2008). A skew-tolerant strategy and confidence measure for k-NN classification of online handwritten characters. Tech. rep., HP Laboratories. 48

SALTON, G. & BUCKLEY, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, **24**, 513–523. 17, 18

SALTON, G., WONG, A. & YANG, C.S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, **18**, 613–620. 16

SAUNIER, N., MIDENET, S. & GRUMBACH, A. (2004). Stream-based learning through data selection in a road safety application. In E. Onaindia & S. Staab, eds., *Proceedings of the Second Starting AI Researchers' Symposium, STAIRS '04*, vol. 109 of *Frontiers in Artificial Intelligence and Applications*, 107–117, IOS Press, Valencia, Spain. 35

Schapire, R.E. & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, **37**, 80–91. 47

Schapire, R.E. & Singer, Y. (2000). Boostexter: A boosting-based system for text categorization. *Machine Learning*, **39(2/3)**, 135–168. 19

Schein, A.I. & Ungar., L.H. (2005). Active learning for multi-class logistic regression. In *Learning 2005 Workshop*, Snowbird, Utah. 58

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. 32

Schneider, K. (2003). A comparison of event models for naive bayes anti-spam e-mail filtering. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1*, 307–314, Association for Computational Linguistics, Budapest, Hungary. 25

Schneider, K. (2004). On word frequency information and negative evidence in naive bayes text classification. In *Advances in Natural Language Processing*, vol. 3230, 474–485, Springer Berlin Heidelberg, Berlin, Heidelberg. 25

Schohn, G. & Cohn, D. (2000). Less is more: Active learning with support vector machines. In *Proceedings of the 17th International Conference on Machine Learning*, 839–846, Morgan Kaufmann, San Francisco, CA. 36, 38, 45, 58

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*, 662–671, ACM, Arlington, Virginia, USA. 25, 59

Sculley, D. (2007). Online active learning methods for fast label-efficient spam filtering. In *Proceedings fo the Fourth Conference on Email and Anti-Spam*, Mountain View, California USA. 33, 35

Seeger, M. (2001). Learning with labeled and unlabeled data. Tech. rep., University of Edinburgh. 69

Segal, R., Markowitz, T. & Arnold, W. (2006). Fast uncertainty sampling for labeling large e-mail corpora. In *Proceedings of the Third Conference on Email and Anti-Spam, CEAS '06*. 33, 40, 41, 43, 59

Settles, B. (2009). Active learning literature survey. Tech. rep., University of Wisconsin–Madison. 32

Settles, B. & Craven, M. (2008). An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1069–1078, ACL Press. 53, 118

Settles, B., Craven, M. & Friedland, L. (2008). Active learning with real annotation costs. In *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*, 1–10. 72

Shen, D., Zhang, J., Su, J., Zhou, G.D. & Tan, C.L. (2004). Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42nd*

Annual Meeting on Association for Computational Linguistics, 589–596, Association for Computational Linguistics, Barcelona, Spain. 33, 56, 109

Shen, X.H. & Zhai, C.X. (2005). Active feedback in ad hoc information retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 59–66, ACM, Salvador, Brazil. 55

Simoff, S.J., Böhlen, M.H. & Mazeika, A. (2008). *Visual Data Mining - Theory, Techniques and Tools for Visual Analytics*, vol. 4404 of *Lecture Notes in Computer Science*. Springer. 60

Singh, M., Curran, E. & Cunningham, P. (2008). Active learning for multi-label image annotation. In *Proceedings of the 19th Irish Conference on Artificial Intelligence and Cognitive Science, AICS '08*. 73

Singh, R., Palmer, N., Gifford, D., Berger, B. & Bar-Joseph, Z. (2005). Active learning for sampling in time-series experiments with application to gene expression analysis. In *Proceedings of the 22nd international conference on Machine learning*, 832–839, ACM, Bonn, Germany. 33

Sohn, S., Kim, W., Comeau, D.C. & Wilbur, W.J. (2008). Optimal training sets for bayesian prediction of mesh assignment. *Journal of the American Medical Informatics Association*, **15**, 546–553. 33

Soucy, P. & Mineau, G.W. (2001). A simple KNN algorithm for text categorization. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, 647–648. 15

Soucy, P. & Mineau, G.W. (2005). Beyond TFIDF weighting for text categorization in the vector space models. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI '05*, 1130–1135, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 17

Su, T. & Dy, J. (2004). A deterministic method for initializing K-Means clustering. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, 784–786, IEEE Computer Society. 199

Sugiyama, M. & Rubens, N. (2008). A batch ensemble approach to active learning with model selection. *Neural Networks*, **21**, 1278–1286. 71

Tang, M., Luo, X. & Roukos, S. (2002). Active learning for statistical natural language parsing. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 120–127, Association for Computational Linguistics, Philadelphia, Pennsylvania. 51

Teixeira, I., de Carvalho, F., Ramalho, G. & Corruble, V. (2002). ActiveCP: a method for speeding up user preferences acquisition in collaborative filtering systems. *Brazilian Symposium on Artificial Intelligence*, **2507**, 237–247. 34, 46

Thiel, K., Dill, F., Kotter, T. & Berthold, M. (2007). Towards visual exploration of topic shifts. In *Proceedings of the 2007 IEEE International Conference on Systems, Man and Cybernetics*, 522–527. 61

Thompson, C.A., Califf, M.E. & Mooney, R.J. (1999). Active learning for natural language parsing and information extraction. In *Proceedings of the Sixteenth International Conference on Machine Learning*, 406–414, Morgan Kaufmann Publishers Inc. 38

Tomanek, K. (2010). *Resource-aware annotation through active learning*. Ph.D. thesis, Technical University of Dortmund. 4, 22, 64

Tomanek, K. & Hahn, U. (2009). Semi-supervised active learning for sequence labeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language*, vol. 2, 1039–1047, Association for Computational Linguistics, Suntec, Singapore. 32

Tomanek, K. & Morik, K. (2010). Inspecting sample reusability for active learning. In I. Guyon, G. Cawley, G. Dror, V. Lemaire & A. Statnikov, eds., *Proceedings of AISTATS 2010 Workshop on Active Learning and Experimental Design*, vol. 16, 169–181, Sardinia, Italy. 66, 67, 68, 150, 152, 161, 163, 165

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*, 45–48, Association for Computational Linguistics, Boulder, Colorado. 64

Tomanek, K., Wermter, J. & Hahn, U. (2007). An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, vol. 3, 486–495, ACL. 65, 67

Tong, S. (2001). *Active Learning: Theory and applications*. Ph.D. thesis, Computer science department, Stanford University. 32

Tong, S. & Chang, E. (2001). Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international conference on Multimedia*, 107–118, ACM, Ottawa, Canada. 32

Tong, S. & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, **2**, 45–66. 36, 40, 41, 44, 47, 55, 58, 64, 152

Tur, G., Hakkani-Tür, D. & Schapire, R.E. (2005). Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, **45**, 171–186. 33

Turtinen, M. & Pietikäinen, M. (2005). Labeling of textured data with co-training and active learning. In *Proceedings of the 4th International Workshop on Texture Analysis and Synthesis*, 137–142. 33, 62

van der Maaten, L. & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, **9**, 2579–2605. 60

van Rijsbergen, C.J. (1979). *Information Retrieval*. Butterworth. 16

Vapnik, V.N. (1995). *The Nature of Statistical Learning Theory*. Springer, 2nd edn. 18, 21

Vidhya.K.A & G.Aghila (2010). A survey of naïve bayes machine learning approach in text document classification. *International Journal of Computer Science and Information Security*, **7**, 206–211. 15, 18, 24

Vlachos, A. (2008). A stopping criterion for active learning. *Computer Speech and Language*, **22**, 295–312. 58

Voorhees, E.M. (1986). *The effectiveness and efficiency of agglomerative hierarchic clustering in document retrieval*. Ph.D. thesis, Cornell University. 203

Vovk, V., Gammerman, A. & Saunders, C. (1999). Machine-Learning applications of algorithmic randomness. *Proceedings of the 16th International Conference on Machine Learning*, 444–453. 48

Wang, K., Zhang, J., Li, D., Zhang, X. & Guo, T. (2008). Adaptive affinity propagation clustering. *0805.1096*, k. Wang, J. Zhang, D. Li, X. Zhang and T. Guo. Adaptive Affinity Propagation Clustering. Acta Automatica Sinica, 33(12):1242-1246, 2007. 145

Wang, M., Hua, X., Song, Y., Tang, J. & Dai, L. (2007). Multi-Concept Multi-Modality active learning for interactive video annotation. In *Proceedings of the International Conference on Semantic Computing, ICSC '07*, 321–328, IEEE Computer Society, Washington, DC, USA. 56

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. 33

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, **1**, 83, 80. 29

Wiratunga, N., Craw, S. & Massie, S. (2003). Index driven selective sampling for cbr. In *Proceedings of the 15th International Conference on Case-Based Reasoning*. 51

Wiratunga, N., Lothian, R. & Massie, S. (2006). Unsupervised feature selection for text data. *Lecture notes in computer science*, 340–354. 15

Xu, Z. & Akella, R. (2008a). Active relevance feedback for difficult queries. In *Proceeding of the 17th ACM conference on Information and knowledge management*, 459–468, ACM, Napa Valley, California, USA. 56

Xu, Z. & Akella, R. (2008b). A bayesian logistic regression model for active relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 227–234, ACM, Singapore, Singapore. 55

Xu, Z., Yu, K., Tresp, V., Xu, X. & Wang, J. (2003). Representative sampling for text classification using support vector machines. In *Proceedings of the 25th*

European conference on IR research, ECIR'03, 393–407, Springer-Verlag, Berlin, Heidelberg. 45, 51, 52

Xu, Z.B., Akella, R. & Zhang, Y. (2007). Incorporating diversity and density in active learning for relevance feedback. In *Proceedings of the 29th European conference on IR research*, ECIR'07, 246–257, Springer-Verlag, Berlin, Heidelberg. 32, 34, 56

Yan, R., Yang, J. & Hauptmann, A. (2003). Automatically labeling video data using multi-class active learning. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, ICCV '03, 516–523, IEEE Computer Society, Washington, DC, USA. 34, 73

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval*, **1**, 69–90. 14

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*, 42–49, Berkley. 15, 16, 21, 29

Yang, Y. & Pedersen, J.O. (1997). A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 412–420, Morgan Kaufmann Publishers Inc. 15

Yao, Z. & Ruzzo, W.L. (2006). A regression-based k nearest neighbor algorithm for gene function prediction from heterogeneous data. *BMC Bioinformatics*, **7**. 48

Yu, D., Varadarajan, B., Deng, L. & Acero, A. (2009). Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion. *Computer Speech and Language*, **24**, 433–444. 34, 71

Zhang, C., Member, S. & Chen, T. (2002). An active learning framework for content-based information retrieval. *IEEE Transactions on Multimedia*, **4**, 260–268. 53

Zhang, Q., Hu, R., Mac Namee, B. & Delany, S.J. (2008). Back to the future: Knowledge light case base cookery. In M. Schaaf, ed., *Proceedings of the 9th European Conference on Case-based Reasoning Workshop on the Computer Cooking Contest*, 239–248. 10, 174, 175, 177

Zhu, J. & Hovy, E. (2007). Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 783–790. 32, 33

Zhu, J., Wang, H. & Hovy, E. (2008a). Learning a stopping criterion for active learning for word sense disambiguation and text classification. In *Proceedings of the Third International Joint Conference on Natural Language Processing*, 366–372. 43

Zhu, J., Wang, H. & Tsou, B. (2008b). 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, 1137–1144. 38, 39, 53, 54

Zhu, J., Wang, H. & Tsou, B. (2009). A Density-Based re-ranking technique for active learning for data annotations. In *Proceedings of the 22nd International Conference on Computer Processing of Oriental Languages*, ICCPOL '09, 1–10, Springer-Verlag, Berlin, Heidelberg. 53

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. 58

Zhu, X., Zhang, P., Lin, X. & Shi, Y. (2007). Active learning from data streams. In *Proceedings of the Seventh IEEE International Conference on Data Mining, ICDM '07*, 757–762, Omaha, NE, USA. 71

Zhu, X.J. (2005). Semi-Supervised learning literature survey. Tech. rep., University of Wisconsin –Madison. 69