Technical Report

UCAM-CL-TR-721
ISSN 1476-2986

Number 721

UNIVERSITY OF
CAMBRIDGE

Computer Laboratory

Investigating classification for natural language processing tasks

Ben W. Medlock

June 2008

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted September 2007 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Fitzwilliam College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Abstract

This report investigates the application of classification techniques to four natural language processing (NLP) tasks. The classification paradigm falls within the family of statistical and machine learning (ML) methods and consists of a framework within which a mechanical 'learner' induces a functional mapping between elements drawn from a particular sample space and a set of designated target classes. It is applicable to a wide range of NLP problems and has met with a great deal of success due to its flexibility and firm theoretical foundations.

The first task we investigate, *topic classification*, is firmly established within the NLP/ML communities as a benchmark application for classification research. Our aim is to arrive at a deeper understanding of how class *granularity* affects classification accuracy and to assess the impact of representational issues on different classification models. Our second task, *content-based spam filtering*, is a highly topical application for classification techniques due to the ever-worsening problem of unsolicited email. We assemble a new corpus and formulate a state-of-the-art classifier based on structured language model components. Thirdly, we introduce the problem of *anonymisation*, which has received little attention to date within the NLP community. We define the task in terms of obfuscating potentially sensitive references to real world entities and present a new publicly-available benchmark corpus. We explore the implications of the subjective nature of the problem and present an interactive model for anonymising large quantities of data based on syntactic analysis and active learning. Finally, we investigate the task of *hedge classification*, a relatively new application which is currently of growing interest due to the expansion of research into the application of NLP techniques to scientific literature for information extraction. A high level of annotation agreement is obtained using new guidelines and a new benchmark corpus is made publicly available. As part of our investigation, we develop a probabilistic model for training data acquisition within a semi-supervised learning framework which is explored both theoretically and experimentally.

Throughout the report, many common themes of fundamental importance to classification for NLP are addressed, including sample representation, performance evaluation, learning model selection, linguistically-motivated feature engineering, corpus construction and real-world application.

Acknowledgements

Firstly I would like to thank my supervisor Ted Briscoe for overseeing the progress of this doctorate; for his relentless belief in the quality of my work in the face of what often seemed to me highly convincing evidence to the contrary. I would also like to thank my colleagues here in the Lab for their companionship, and for fruitful discussion and advice; in particular, Andreas Vlachos and Mark Craven who both spent time reading my work and offering invaluable technical guidance, and Nikiforos Karamanis who generously helped with annotation and discussion. At different times and places, the following people have all contributed to this work by offering me the benefit of their expertise, for which I am very grateful: Karen Spärck Jones, Ann Copestake, Simone Teufel, Joshua Goodman, Zoubin Ghahramani, Sean Holden, Paul Callaghan, Vidura Seneviratne, Mark Gales, Thomas Hain, Bill Hollingsworth and Ruth Seal. This research was made possible by a University of Cambridge Millennium Scholarship.

On a personal note, I would like to acknowledge my sirenical office partners Anna and Rebecca; it has been a pleasure to begin and end my postgraduate journey with you. I would also like to thank my PhD buddies Wayne Coppins and Ed Zaayman for making the ride so much more entertaining, and in fact all my friends, both in and out of Cambridge; you're all part of this. Special acknowledgement goes to my grandfather George who kept me on my toes with his persistent enquiry into my conference attendance and publication record, and to Lizzie for proofreading and patiently allowing me to defend the split infinitive. Final and profound thanks goes to my family: Paul, Jane, Matt and Abi, for everything you mean to me, and to my Creator who fills all in all.

For George, who did it all on a typewriter...

Contents

Chapter 1

Introduction

Natural Language Processing (NLP) is the application of computational models to tasks involving human language text. NLP research has been active since the dawn of the modern computational age in the early 1950s, but the field has burgeoned in recent years, fueled by the rapid development of the internet and consequent increase in the availability of online text. Initially, symbolic approaches to NLP tasks were heavily pursued, drawing on research from the linguistics and propositional logic communities, but in recent years, statistical and machine learning (ML) methods have become increasingly dominant.

An important problem-solving paradigm falling within the family of statistical and ML methods is *classification* where a mechanical 'learner' induces a functional mapping between elements drawn from a particular sample space and a set of designated target classes. The classification paradigm is broadly applicable to a wide range of NLP problems and has met with a great deal of success due to its flexibility and firmly established theoretical foundations.

In this report we investigate the application of classification methods to four topical NLP tasks. Our investigation covers a wide range of techniques, focusing predominantly, though not exclusively, on probabilistic approaches, and exploring not only the classification models themselves but also related issues such as data annotation, sample representation and evaluation strategies. In some cases we investigate tasks that are relatively new to the NLP community, while in others we explore better-known tasks, seeking a deeper understanding of both the problems and proposed solutions. Our focus is on analysis of the *behaviour* of classification models as they are applied to each of the NLP tasks, rather than on theoretical mathematical analysis of the models themselves, though some understanding of statistics and probability theory will be necessary. Our aim is to investigate why different methods are appropriate for particular NLP tasks, as well as demonstrating their effectiveness through experiment.

1.1 Project Overview

Four NLP classification tasks are investigated in this report; here we provide a brief introduction to each of them.

1.1.1 Topic Classification

Topic classification is the task of assigning topic labels to textual documents. It is an instance of the more general 'text categorization' paradigm and has become something of a benchmark problem within the NLP/ML communities due to the simplicity and clarity of its formulation, the relative abundance of manually classified online text amenable to experimentation and the many evident real-world applications for topic classification systems.

1.1.2 Spam Filtering

Interest in spam filtering as a classification task has burgeoned over recent years in response to the ever-worsening spam problem. We examine the problem of *content-based* spam filtering, where classification is performed on the basis of the content of the email message rather than on meta-level information such as the domain of origin or the number of recipients. Most content-based approaches to spam filtering frame the problem as a text categorization task, given that email almost always contains some form of textual content. Various aspects of the spam filtering problem set it apart from more traditional classification tasks (such as topic classification). For example spam filters are often evaluated under the *asymmetric misclassification cost* paradigm, due to the fact that mislabeling a genuine email is usually much more serious than mislabeling an item of spam.

1.1.3 Anonymisation

In our context, anonymisation is the process of obscuring sensitive references within a body of text, thus allowing it to be shared for research or other purposes. The task involves identifying and classifying references and has thus far been given little attention within the NLP community, though our formulation of the task is quite similar to the relatively well-studied problem of named entity recognition (NER). As a classification problem, anonymisation is significantly finer-grained than either topic classification or spam filtering, operating within the sample space of individual references (single terms or short phrases) – usually no more than a few words long – rather than whole documents. The nature of the anonymisation task raises various issues relating to how an NLP-aided anonymisation system would be deployed and the type of constraints that would limit its practical utility. We discuss such issues in our investigation.

1.1.4 Hedge Classification

Hedge classification is the task of automatically labelling sections of written text according to whether or not they contain expressions of linguistic 'affect' used by the author to convey speculativity. It is a relatively new task within the NLP community and is currently of growing interest due to the expansion of research into the application of NLP techniques to scientific literature from the fields of biomedicine/genetics (*bioinformatics*). NLP techniques are often used to identify and extract experimental results reported in scientific literature and thus rely on the veracity of the reported findings. However, experimental conclusions are often hedged by the authors if they consider them to be only speculative or potentially unreliable, and it is useful to be able to automatically identify when this is the case. In our formulation, the task involves classification at the sentence level, and therefore differs in granularity from the other tasks.

1.2 Research Questions

We seek to address a number of key research questions in the course of this study; some are addressed specifically in certain chapters, others by drawing conclusions from multiple experimental analyses. The questions are outlined here, and later (§7.1) we assess how effectively they have been / can be answered.

1. Is category separability a reliable correlate for classification accuracy and can it be used as a guide to classifier selection?

2. In applied NLP/classification, is it more important to focus on the sample representation or the machine learning model?

3. Are linguistically-motivated features, especially those derived from a 'deep' syntactic analysis of text, useful for classification?

4. The general consensus of previous research is that linguistically motivated features are not useful for classification. If our work corroborates this view, why is this the case?

5. Is there a correlation between sample resolution and the utility of complex features?[1] If so, why?

6. Can the task of textual anonymisation be formulated in a fashion that is both amenable to NLP technologies and useful for practical purposes?

7. Can the task of sentence-level hedge classification be specified so as to achieve high agreement amongst independent annotators?

8. Can a high level of accuracy be achieved on the hedge classification task using semi-supervised machine learning?

Specific Goals

Some of the specific aims of this project are:

- Develop a method for quantifying category separability (a measure of the 'distance' between category distributions – §3.1) and examine its impact on classification accuracy.

- Explore the use of linguistically motivated features in fine grained topic classification.

- Develop a new state-of-the-art content-based spam filtering model, taking account of the semi-structured nature of email.

[1] 'Complex' features are those that consist of combinations of single term-based features.

- Construct a new anonymised spam filtering corpus with a sizeable proportion of heterogeneous genuine email messages.
- Present the problem of anonymisation as a reference level classification task.
- Develop annotation schemes and a publicly available corpus of informal text to facilitate anonymisation experiments.
- Develop an interactive learning model for anonymisation, motivated by the subjectivity of the task.
- Specify the problem of hedge classification as a sentence-level classification task.
- Develop new annotation guidelines for identifying and labeling hedging in scientific literature.
- Assemble a sizeable corpus of biomedical text sentences for hedge classification.
- Investigate the properties of the hedge classification task from a semi-supervised machine learning perspective.
- Develop and explore a probabilistic model for training data acquisition.

1.3 Tools

Throughout our work we make use of various external machine learning and NLP tools; here we introduce two of the more ubiquitous ones.

1.3.1 RASP

RASP (*Robust Accurate Statistical Parsing*) (Briscoe, Carroll & Watson 2006)[2] is a domain-independent, robust parsing system for English with state-of-the-art performance in benchmark tests (Preiss 2003). At its core is a unification-based context-free grammar (CFG) over part-of-speech tags, with a probabilistic GLR (generalised left-right) parsing engine. The complete system consists of a number of pipelined components, including:

- Sentence splitting
- Part-of-speech tagging
- Morphological analysis
- Parsing

It produces output in various different formats including syntax trees and grammatical relations (GRs). The syntax tree and GR outputs for the sentence: *Every cat chases some dog.* are shown in Figure 1.1 and 1.2 respectively. GRs are labeled, binary relationships between terms, the first term being the 'head' and the second the 'dependent'. For example, the three GRs in Figure 1.2 are as follows:

- *ncsubj*: Non-clausal subject (dependent) – verb (head) relation.
- *dobj*: Verb (head) – direct object (dependent) relation.
- *det*: Determiner (dependent) – nominal (head) relation.

[2] *http://www.informatics.susx.ac.uk/research/nlp/rasp*

```
(|T/txt-sc1/-+|
  (|S/np_vp| (|NP/det_n1| |Every:1_AT1| (|N1/n| |cat:2_NN1|))
   (|V1/v_np| |chase+s:3_VVZ|
    (|NP/det_n1| |some:4_DD| (|N1/n| |dog:5_NN1|))))
  (|End-punct3/-| |.:6_.|))
```

Figure 1.1: RASP syntax tree output - *Every cat chases some dog.*

```
(|ncsubj| |chase+s:3_VVZ| |cat:2_NN1| _)
(|dobj| |chase+s:3_VVZ| |dog:5_NN1|)
(|det| |dog:5_NN1| |some:4_DD|)
(|det| |cat:2_NN1| |Every:1_AT1|)
```

Figure 1.2: RASP GR output - *Every cat chases some dog.*

1.3.2 SVM^{light}

SVM^{light} Joachims (1999)[3] is an implementation of the popular Support Vector Machine (SVM) classification framework (Vapnik 1995). It is relatively efficient and has been shown to yield state-of-the-art performance on a variety of NLP tasks. SVM optimization is quadratic in the number of training instances, and SVM^{light} makes the problem tractable by decomposing it into small constituent subproblems (the 'working set') and solving these sequentially. The popularity of SVM^{light} stems from its efficiency (especially when using the linear kernel) and its ease of use. Extensions to the basic package include a version that can be applied to structured data (SVM^{struct}), and also more recently a version with claimed linear optimization complexity (Joachims 2006).

1.4 Report Structure

The structure of the report is as follows: chapter 2 provides an introduction to the classification paradigm and tells the story of its development within NLP, highlighting various important contributions from the literature. The aim here is to provide the reader with a broad understanding both of the classification framework and of its application to NLP. Chapters 3 to 6 address the previously introduced classification tasks – topic classification, spam filtering, anonymisation and hedge classification respectively. Finally, chapter 7 concludes with an examination of the contributions of our work and speculation about the direction of future research in the field.

[3] *http://svmlight.joachims.org/*

Chapter 2

Background and Literature Overview

2.1 The Classification Model

According to the OED, *classification* is defined as:

> *The action of classifying or arranging in classes, according to common characteristics or affinities; assignment to the proper class.*[1]

In our context, a class is defined as a (possibly infinite) set of objects, referenced by a unique class label which is an arbitrary descriptor for the set. The classification task, then, is to assign class labels to objects. In reality, objects are represented by sets of measurements taken on various aspects of form, structure and content, and the range of possible values of these collective measurements is referred to as the 'sample space'. As an abstract example, consider the task of classifying earthenware jugs into historical categories.[2] The 'measurements' chosen to represent each jug might include size, volume, glaze type, number of handles, etc. The sample space then consists of all the possible values of these measurements, where each measurement type represents a 'dimension' in the sample space.

At this point it is necessary to introduce some notation:

$\mathcal{X}$	sample space
$\mathcal{Y}$	set of class labels
$\mathbf{x} \in \mathcal{X}$	sample – vector of measurements representing a particular object
$y \in \mathcal{Y}$	class label

Given these concepts, the classification task is to induce a functional mapping between objects in the sample space and class labels, the 'classification function', taking the form:

$$f : \mathcal{X} \to \mathcal{Y} \tag{2.1}$$

This gives rise to a scenario in which each object is associated with only one class label (e.g. a member of a single class set). There are of course many conceivable scenarios in which this is not the case; for example a news story about a football club takeover bid could belong both to the 'sport' class and also to the 'finance' class. If we are to allow

[1] *The Oxford English Dictionary.* 2nd ed. 1989. OED Online. Oxford University Press. 4 Apr. 2000

[2] We will use the terms 'class' and 'category' interchangeably.

Table 2.1: Classification model types

Domain ($\mathcal{X}$)	**Range** ($\mathcal{Y}$)	**Function**	**Classification type**
Fruit	{Ripe, Unripe}	$f : \mathcal{X} \to \mathcal{Y}$	Single-label, binary (2-class)
Fruit	{Apple, Orange, Pear}	$f : \mathcal{X} \to \mathcal{Y}$	Single-label, multi-class
People	{Tall, Handsome}	$f : \mathcal{X} \to \mathcal{PY}$	Multi-label, binary
People	{Tall, Dark, Handsome}	$f : \mathcal{X} \to \mathcal{PY}$	Multi-label, multi-class

multiple class labels per sample, we must modify the classification function to represent mapping of samples to *sets* of class labels:

$$f : \mathcal{X} \to \mathcal{PY} \tag{2.2}$$

where $\mathcal{P}$ denotes 'powerset'. This is sometimes referred to as *multi-label* classification, not to be confused with *multi-class* classification which refers to the scenario in which the number of categories in a given problem is greater than two (the two class problem is referred to as *binary* classification). Examples of these different classification model types are given in table 2.1. In multi-label classification, the set of class label assignments can be augmented with numerical values representing the degree, or probability, of membership for each assigned class. This is sometimes referred to as 'soft' classification, as opposed to 'hard classification' where class membership is strictly binary (yes/no). The classification tasks examined in this report are all single-label.

2.1.1 Classification by Machine Learning

There are many methods of constructing classification functions. For instance, a trained police dog learns a function that maps physical substances to the classes 'legal' or 'illegal'. However, we are concerned with a particular method of inducing the classification function – by inductive *machine learning* (ML). This means that the parameters governing the detailed behaviour of the classification function are 'learned' through a mechanical training procedure which can be formalised in the language of mathematics. Machine learning is usually considered a broad subfield of *artificial intelligence* (AI) and has grown in scope and popularity over recent years. It is closely related to fields such as *data mining* and *statistics* but has increasingly taken on its own identity, consisting of a unique blend of statistical learning theory, applied probabilistic methods and theoretical computer science.

2.1.2 Classification versus Clustering

It is only relatively recently that classification has emerged as properly distinct from the related task of *clustering.* The two are closely linked, and many of the techniques used in classification are also applicable to clustering. The distinction between the tasks, however, is that in classification, class identity is known before the learning procedure begins, whereas in clustering, classes (clusters) are induced during the learning process. More formally, the classification task is characterised by the existence of a theoretical 'target function', t, of the same form as the classification function, such that for any given sample $\mathbf{x}$, $t(\mathbf{x})$ returns the label(s) representing the class membership of $\mathbf{x}$. The goal of the learning procedure, then, is to approximate the target function. Conversely, in clustering there is no target function; rather, class structure is learned by grouping samples into clusters based on some form of *similarity metric*.

2.1.3 Supervision

In practice, the existence of the target function for a classification task is made manifest through a set of training samples of the form $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where it is assumed that each training sample $(\mathbf{x}_i, y_i)$ reveals the value of the target function at that point in sample space, e.g. $t(\mathbf{x}_i) = y_i$. The training data is used in the learning process to approximate the true target function, and this is referred to in the machine learning literature as *supervised learning*. Conversely, clustering is an example of an *unsupervised* learning task, because there is no target function, and therefore no requirement for supervision (training data). There are various forms, or 'strengths' of supervised learning; for instance, *semi-supervised learning* refers to the scenario in which there is initially only a very limited amount of labelled training data, and 'bootstrapping'[3] is employed to utilise additional unlabelled data.

2.1.4 Approaches to Classification

We now consider, at a high level, some of the different machine learning approaches that have been used to tackle the classification problem.

Probabilistic

The laws of probability theory provide an expressive, theoretically justifiable framework within which the classification problem can be explored. For an arbitrary point in sample space, $\mathbf{x}_k$, the target function $t(\mathbf{x}_k)$ returns the true class membership for the object represented by $\mathbf{x}_k$. However, without access to the target function, a sensible approach is to encode *uncertainty* about the value of the target function by estimating the *probability* of class membership across the sample space. Formally, for any arbitrary input $\mathbf{x}$ and class label y, we seek to estimate $P(y|\mathbf{x})$, known as the *posterior class probability.* This can be interpreted as the conditional probability of the class given the sample. Subsequently, *decision theory* can be used to predict class membership in a such a way that the *classification error* is minimized. See Bishop (2006) for a more detailed discussion of these concepts.

One approach to estimating $P(y|\mathbf{x})$ is to decompose it using *Bayes' Rule* in the following manner:

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y) \cdot P(y)}{P(\mathbf{x})} \tag{2.3}$$

The denominator, $P(\mathbf{x})$, can either be ignored as it is invariant to class, or marginalised:

$$P(\mathbf{x}) = \sum_i P(y_i) \cdot P(\mathbf{x}|y_i)$$

[3] Bootstrapping is the process of using a small amount of labeled training data to induce further labels, which are then used as additional training samples and the process is iterated up to a specified termination point.

The remaining quantities to be estimated are $P(y)$, the *class prior* and $P(\mathbf{x}|y)$, the *class conditional likelihood* or *density*. Techniques that model the input distribution through estimation of the (conditional or unconditional) sample density are known as *generative* because in theory it is possible to use these estimates to 'generate' synthetic samples. Examples of generative classifiers include *Naïve Bayes* (Maron 1961, Duda & Hart 1973, Fuhr 1989, McCallum & Nigam 1998, Rennie, Shih, Teevan & Karger 2003) and *Maximum Entropy* (MaxEnt) (Jaynes 1957, Berger, Pietra & Pietra 1996, Malouf 2002).

Alternatively, the posterior class probabilities can be estimated directly, in which case the model parameters are chosen such that the probability of correctly predicting the training data is maximised. Approaches that model the posterior probabilities directly are known as *discriminative*. Discrimintative probabilistic classifiers include those that fall within the *logistic regression* family (Fuhr & Pfeifer 1994, Zhang & Oles 2001, Zhang, Jin, Yang & Hauptmann 2003, Genkin, Lewis & Madigan 2005).

In recent years it has become popular within certain branches of the machine learning community to adopt an approach which consistently applies the laws of Bayesian probability theory to all stages of machine learning inference, sometimes called *full Bayesian treatment*. Such approaches are quite powerful as they allow the mechanics of Bayesian inference to induce optimal model structure from the data, given the prior distributions. Minimal constraints are imposed on the structural properties of the model, allowing, in principle, a closer approximation to the true distribution from which the data are drawn. Disadvantageously, they also tend to be rather costly from a computational perspective, as they require (at least in theory) integration over the space of all models. In practice, sampling methods are often used, such as variants of *Markov Chain Monte Carlo* (MCMC), or approximate inference techniques such as *expectation propagation*. For further details the reader is referred to the following: Ghahramani (2005), Rasmussen & Williams (2005), Bishop (2006).

Non-probabilistic

There are also many successful classification techniques based on non-probabilistic models. Such approaches usually attempt to model the classification function directly. There are many well-studied techniques for functional estimation within the statistics literature that have found application in the area of machine learning in general and classification in particular. A good example of a non-probabilistic framework for machine learning and classification is *structural risk minimization* (SRM) (Vapnik & Chervonenkis 1974, Vapnik 1995), which provides the theoretical motivation for classification models such as the popular *support vector machine* (SVM). SVMs fall within a family of classifiers known as *kernel machines* which make use of the *kernel trick*, allowing high dimensional non-linear problems to be solved by linear optimization methods through a particular type of input space mapping called the *kernel function*. See Scholkopf & Smola (2001) for further details. Note that any classifier yielding real-valued numerical output representing some notion of prediction 'confidence' or 'margin' can be transformed into a discriminative probabilistic model using a mapping function, for instance of the sigmoidal family. Alternative non-probabilistic classification methods include *decision trees* (Fuhr & Buckley 1991, Quinlan 1993, Weiss, Apte, Damerau, Johnson, Oles, Goetz & Hampp 1999), memory-based learning methods such as *k-nearest neighbour* (Masand, Linoff & Waltz 1992, Lam & Ho 1998) and variants of the *perceptron algorithm* (Rosenblatt 1958, Krauth & Mezard 1987, Freund

& Schapire 1998, Bruckner & Dilger 2005).

Sequential

When the samples in a given classification domain are ordered sequentially with strong interdependencies, as is the case for instance in part-of-speech tagging for natural language, or phoneme labeling in continuous speech recognition, a somewhat different classification paradigm is often used to the one we have presented so far. Models that are specifically designed to deal with sequential data are known as *sequential classifiers* and can also be designated within the generative/discriminative probabilistic/non-probabilistic framework. A popular generative probabilistic sequential classifier is the *Hidden Markov Model* (HMM) which utilises the *Markov assumption*[4] to label sequence data efficiently (Rabiner 1989, Cappé, Moulines & Ryden 2005). More recently *Conditional Random Fields* (CRFs) have been introduced with the ability to condition the category of the focus sample on features of arbitrary dependency distance (Lafferty, McCallum & Pereira 2001, Sutton & McCallum 2006). In our work we focus almost exclusively on non-sequential classification models.

2.1.5 Generalisation

One of the central topics in machine learning for tasks such as classification is *generalisation.* Put simply, correct prediction of the training data does not guarantee good accuracy on unseen data. The problem is that for classification problems of reasonable complexity there is always a large (usually infinite) number of functions which correctly (or almost correctly) predict the training data, and most of them will not closely approximate the target function. In general, if the classification model is too complex (too many degrees of freedom) it will learn a function that *over-fits* the training data and fails to generalise. Conversely, if it is too simple it will not have the requisite flexibility to perform well on either the training or the unseen data. Recent advances in machine learning have shed a good deal of light on this topic, and most modern approaches to classification utilise techniques for restricting model complexity, whilst maintaining low training error. Once again, we refer the reader to Bishop (2006) for a full treatment of this, and related topics.

2.1.6 Representation

A key issue in constructing classification systems is *representation.* Using the terminology introduced earlier, the type of measurements that are taken on a collection of objects will have an enormous impact on how successfully they can be classified. To return to the earthenware jugs example, if measurements are taken of just the physical dimensions of each jug, this will clearly limit the effectiveness of any classification procedure; it is necessary also to take measurements revealing something of the form and style of each item. In general it is beneficial to take measurements of a type that will give purchase on those aspects of each object that are relevant to the classification task. Within the NLP/ML communities, the process of choosing how to represent objects is called

[4] The Markov assumption states that the category of the current sample is independent of all but the current and previous model states, thereby greatly decreasing inference complexity.

feature generation and *feature selection*. The former refers to the task of deciding what type of measurements to make, and the latter to the process of choosing which of these measurements will actually be useful in solving the problem. For example, in the task of document classification, the feature generation step might be to extract single terms as features, while the feature selection step would be to decide which terms (if not all) will reveal the aspects of each document representative of the intended class structure.

There are two broad philosophies when approaching the problem of representation from a machine learning perspective. The first is to use feature selection techniques to construct representations that are sparse and tailored to the particular problem, thus protecting the classifier against overfitting and improving accuracy. The second is to generate a large number of features, perform only minimal feature selection, and then employ classification models that automatically 'select' relevant features as an outcome of their mathematical properties. The advantage of the first approach is that it facilitates the use of relatively simple classification models that are easy to implement and highly efficient. Disadvantageously, there is no principled method for setting thresholds on feature relevance, rather the number of features to retain must be chosen on the basis of performance tuning using techniques such as *cross validation*. The second approach requires slower, more complex classification models, and a certain level of perspicuity is often lost in terms of the effect of particular features on classification accuracy. However, problematic thresholding decisions are avoided, and the overall accuracy of this approach is often marginally superior.

At this point it is also worth noting that there is often a discrepancy of perspective between the focus of theoretical academic research on ML classification and that of industry with respect to the issue of feature engineering versus model engineering. Academic research tends to focus on development of the classification models, where there is a great deal of inherent interest in the theoretical properties of the models themselves. Different methods are then evaluated on modest-sized datasets and small but statistically significant improvements in performance are deemed genuinely significant. On the other hand, industrial researchers often have access to extremely large datasets and tend to focus more on identifying and extracting highly representative/discriminative feature sets, after which the choice of classification model has relatively little impact on overall accuracy. The work presented in this report sits somewhere between these two perspectives. Academic investigation into applied NLP is concerned with examining the theoretical properties of statistical and ML techniques with respect to NLP tasks, while at the same time striving to make a realistic assessment of their performance in real world situations.

2.2 Classification in NLP – Literature Overview

The aim of this section is to provide the reader with an overview of developments that led to the widespread use of the classification paradigm in natural language tasks. Perhaps the earliest reference to the idea of combining statistics with computational models for analysing language is Luhn (1957). The motivation in this work is to move toward constructing a reliable mechanised library search system based on statistical analysis of term usage within the domain of technical literature. It is of interest to note that in contrast to more recent work, the paper focuses a significant amount of attention on addressing philosophical concerns such as how ideas and experiences are distilled into language and communicated, and what an underlying, language-independent, ‘syntax of notions’ might look like. In time, repeated experimentation into the application of statistical techniques led the NLP community to adopt the view that relatively simple ‘surface’ representations almost always yield better results than complex representations of underlying concepts.

The application of classification techniques to tasks falling within NLP grew mainly out of the efforts of the information retrieval (IR) community through the 1960s and 70s. Early IR researchers had the intuition that grouping documents by some notion of class ought to provide reasonable grounds for making relevance judgements. At the same time, it was thought that classifying terms in a similar fashion could likewise be useful for improving models of query-text relevance. This early work on classification within IR is surveyed in inimitable fashion by the late Karen Sparck Jones (1991). She highlights various references considered important from the perspective of classification research in general before focusing on IR in particular. A key point made in the paper is that most of this work treated classification not as an end in itself but as a means of improving retrieval precision/recall, and actually met with relatively little success in comparison to more direct statistical term-weighting retrieval models.

Cormack (1971) surveys pre-1970 classification research, though it is clear that what was understood as classification in this context is more akin to our definition of clustering, the task being framed not in terms of estimating a target function or prior categories, but rather with a strong focus on ‘distance metrics’ for computing similarity between objects. However, many of the concepts and techniques discussed have been adapted for developing classification models in the modern sense, and the paper contains descriptions of many important algorithmic, statistical and information theoretical models. The paper contains many references to research carried out during the 1960s on clustering methods, sourced mainly from the areas of statistics and the natural sciences, where there was a significant interest in developing computational models for automatic, or semi-automatic taxonomy construction.

One of the earliest references to text classification using machine learning is Maron (1961). In fact, at this stage the field of machine learning had not properly emerged in its own right, and the classification task is framed as *automatic indexing* of documents, but the concepts are clearly analogous. Consider the following quote, from the paper:

> *[Our] approach... is a statistical one... based on the rather straightforward notion that the individual words in a document function as clues, on the basis of which a prediction can be made about the subject category to which the document most probably belongs.*

The generalisation of this idea is in some sense the foundation for research into machine learning methods for classification tasks involving natural language. Put simply, the general principle is that terms comprising language units (phrases, sentences, documents etc.) are both amenable to statistical analysis and act as indicators of some sense of ‘semantic category’.

Maron (1961) goes on to present a probabilistic classification model, making use of the *independence assumption* which forms the basis of the highly popular *Naïve Bayes* classifier (though it is not referred to as such in Maron’s work). His experiments make use of a modest dataset of around 500 technical documents, and he addresses various key issues such as feature selection (‘cue word selection’ in his terms) and function word (stopword)

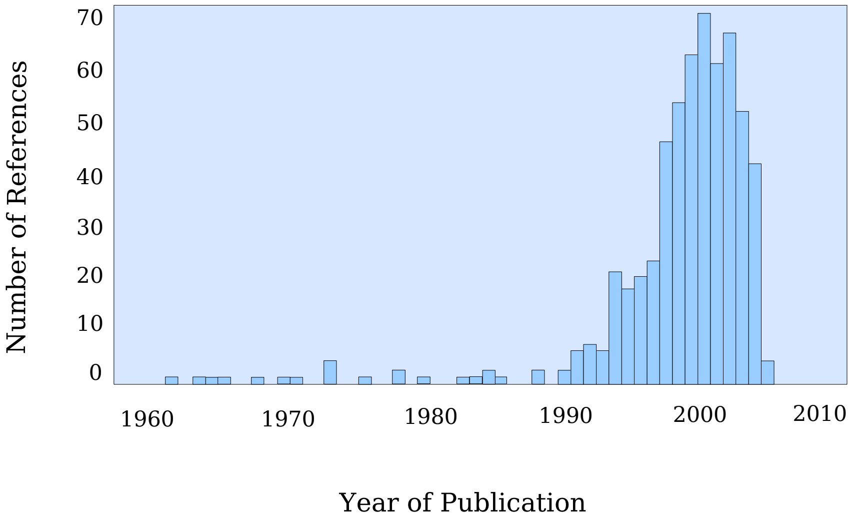


Figure 2.1: Publications on text categorization

filtering. The features are selected manually through analysis of the distribution of words across categories.[5] The outcome of the reported experiments was interpreted as reasonably encouraging, though the limited amount of available data raised inevitable questions about scalability. Overall, Maron's paper is a seminal pioneering investigation, and many of the concepts that are explored in depth in later classification research exist here in embryonic form.

Following this work, a comparative study was carried out a few years later on the same dataset (Borko & Bernick 1963, Borko & Bernick 1964). The method investigated in this work is a variant of linear algebraic *factor analysis*, using the same features as Maron (1961). It was found to perform quite similarly to Maron's probabilistic method, and the main outcome of the study is to reaffirm the feasibility of automatic document classification using computational methods.

Following the precedent set by this early work and the increasing demand for document-based retrieval systems, topic classification (also referred to as topic spotting, document routing or automatic indexing) became the benchmark application for the development and evaluation of classification techniques within the NLP/IR communities. However, due to the limited success of classification methods in early IR, widespread research into classification techniques for tasks such as text categorization was slow to take off, with little more than a handful of articles appearing through the 1970s and 80s on the topic, most in a similar vein to early studies (Field 1975, Hamill & Zamora 1980, Robertson & Harding 1984).

Figure 2.1[6] plots the number of text categorization publications appearing from the early 1960s onwards. Little was published before about 1990, at which point there was a

[5] The concept of automatic feature ranking and selection by information theoretical properties is in fact postulated by Maron, though not implemented.

[6] This data was gathered as part of a project to collate an online public bibliography of text categorization research – http://liinwww.ira.uka.de/bibliography/Ai/automated.text.categorization.html

dramatic increase in research effort, coinciding with burgeoning interest in the relatively new field of machine learning.[7] We refer the reader to Sebastiani (2002) for a comprehensive survey of research into machine learning methods for text categorization. Two important factors that contributed to this rapid expansion were the emergence of benchmark datasets such as the Reuters newswire corpora[8] and MUC (*Message Understanding Conference*) data[9], and concurrently a broad consensus on appropriate evaluation metrics for classification tasks, pioneered by David Lewis and others (Lewis 1991).

In the early 1990s, the emergence of sizeable real-world NLP benchmark corpora presented a new challenge to those investigating the application of machine learning methods. High dimensional, sparse data and large scale datasets resulted in significant theoretical and computational demands, prompting the development of robust and efficient classification models such as decision trees (Fuhr & Buckley 1991, Quinlan 1993), kNN (Masand et al. 1992) and the various forms of Naïve Bayes (Maron 1961, Duda & Hart 1973, Fuhr 1989). The efficiency of models such as Naïve Bayes and kNN allowed early experimental studies into large scale classification, and also issues such as feature selection and sample representation (Masand et al. 1992, Lewis 1992), laying the groundwork for much subsequent research.

Throughout the 1990s topic spotting remained the dominant outlet for classification research within NLP, and the level of understanding and range of approaches increased dramatically, fueled by rapid progression within the ML community in general (Apte, Damerau & Weiss 1994, Fuhr & Pfeifer 1994, Cohen 1995, Damashek 1995, Riloff 1996, Hull, Pedersen & Schütze 1996, Joachims 1997). In the late 1990s the NLP research community was impacted quite significantly by the introduction of kernel- and margin-based techniques and in particular the popularisation of Support Vector Machines (SVMs) (Vapnik 1995) thanks to publicly distributed software packages such as SVM^{light} (Joachims 1999) and LibSVM (Chang & Lin 2001). Joachims (1998) demonstrated the effectiveness of SVMs for text categorization and his results were confirmed in a number of other important publications appearing around the same time (Dumais, Platt, Heckerman & Sahami 1998, Yang & Liu 1999).

Popular, high-performance classification models distributed in freely available implementations such as the SVM packages and general purpose machine learning toolkits like MALLET (McCallum 2002)[10] and WEKA (Witten & Frank 2002)[11], along with an increased understanding of theoretical issues such as classifier generalisation, sample representation and feature selection has also fuelled the application of classification techniques to an ever-widening range of NLP-related applications, including:

- *genre detection* (Kessler, Nunberg & Schütze 1997, Wolters & Kirsten 1999, Rehm 2002, Finn, Kushmerick & Smyth 2002)
- *sentiment analysis* (Pang, Lee & Vaithyanathan 2002, Pang & Lee 2004, Whitelaw, Garg & Argamon 2005)
- *content-based spam filtering* (Sahami, Dumais, Heckerman & Horvitz 1998, Drucker, Wu & Vapnik 1999, Androutsopoulos, Paliouras, Karkaletsis, Sakkis, Spyropoulos

[7] The journal *Machine Learning* was first published in 1986.

[8] *http://www.daviddlewis.com/resources/testcollections*

[9] *http://www.itl.nist.gov/iaui/894.02/related_projects/muc*

[10] *http://mallet.cs.umass.edu/index.php/Main_Page*

[11] *http://www.cs.waikato.ac.nz/ ml/weka*

& Stamatopoulos 2000, Rios & Zha 2004)

- *authorship attribution* (Stamatatos, Fakotakis & Kokkinakis 1999, Diederich, Kindermann, Leopold & Paass 2003)
- *information extraction* (Collins & Singer 1999, Kambhatla 2004, Zhao & Grishman 2005)
- *word sense disambiguation* (Pedersen 2000, Chao & Dyer 2002, Martínez, Agirre & Màrquez 2002)
- *named entity recognition* (Bennett, Aone & Lovell 1997, Borthwick 1999, Shen, Zhang, Su, Zhou & Tan 2004)

The application of classification to NLP continues to expand rapidly and the work in this report contributes to this expansion, both in terms of identifying new tasks to which classification models can be applied, and also by deepening our understanding of existing areas of application. In addition to the background literature survey presented in this chapter, each subsequent chapter contains a more focused presentation of work directly relevant to the task and models in question.

Chapter 3

Topic Classification

In this chapter we examine the problem of *topic classification* using data drawn from the recent Reuters RCV1 newswire corpus (Lewis, Yang, Rose & Li 2004). We demonstrate from experimental analysis that classification performance is positively correlated with category separability and that this can be quantified and potentially used as a guide for classifier selection. We compare popular generative and discriminative classification models and investigate techniques for improving the sample representation by removing noise and balancing the distribution sizes. Our results show that such methods can improve the accuracy of a simple, efficient generative model to make it competitive with its more complex discriminative counterparts on balanced data. We also explore techniques for complex feature (bigram) generation and present and evaluate a novel approach that uses a syntactic parser to identify terms in key grammatical relationships. Finally, we discuss the theoretical complexity and efficiency of the techniques presented in the study.

3.1 Introduction and Motivation

It is well known in the text classification (TC) community that classification performance correlates positively with the amount of available training data (Yang & Liu 1999, Lewis et al. 2004, Debole & Sebastiani 2004). Whilst it is important to assess the effectiveness of classification in the presence of limited training data, this can often become the dominant factor in whole-corpus TC experiments, obscuring other important issues that affect classification performance. This is especially true in experiments involving complex features as such techniques necessarily expand the feature space and can chronically disadvantage a classifier with very limited training data. Some researchers have attempted to circumvent this problem by restricting their experiments to only the most frequent categories of popular TC test corpora (McCallum & Nigam 1998, Dumais et al. 1998, Bennett 2003). Disadvantageously, this tends to mitigate against wide-scale experimentation into classification of the more interesting 'fine-grained' distinctions, as these categories are often sparsely populated with respect to the corpus as a whole.

The recent release by the Reuters corporation of the sizeable ($\sim$800k document) RCV1 newswire corpus (Lewis et al. 2004) has provided opportunity for a whole new range of TC experiments in a more realistic setting than any offered by previous TC corpora, facilitating experimentation across differing levels of granularity, without the problems that arise due to severe data sparsity. In this study we investigate various levels of granularity within RCV1, normalising for the amount of training data so that other factors affecting classification performance can be properly examined. Our main objectives are:

- Examine the correlation between *category separability* and classification accuracy. Given two categories, we use the phrase 'category separability' to refer to some measure of the distance between their respective sample distributions, as estimated from the available training data. Category separability can be seen as the inverse of the more common notion of 'distributional similarity'. We expect categories that are more separable to be easier to classify and thus we hypothesize that category separability should correlate positively with classification accuracy.

- Compare popular implementations of state-of-the-art text classification algorithms, in particular *Support Vector Machines* and *Boosting.*

- Explore techniques for improving the document representation.

3.2 Methodology

In this section we describe the data, classification models and evaluation metrics used in our experiments.

3.2.1 Data

The RCV1 (Reuters Corpus Volume 1) dataset (Lewis et al. 2004) contains a total of just over 800k news stories extracted over a 12-month period in 1996-97. Documents are marked with three hierarchical category codings, *topic*, *industry* and *region* though in this study we use only the *topic* categories. These are arranged into three levels of granularity – coarse, medium and fine grained. For each category we randomly select 1000 documents for training, 500 for development and 500 for testing and final evaluation. Categories that contain fewer than 2000 documents in total are excluded. By selecting the same number of documents from each category we eliminate the effect of the training data size/classification accuracy correlation and are able to properly investigate accuracy as a function of class granularity. Our experimental framework represents an 'idealised' setting for the TC task, where there is sufficient training data for all categories and the training distribution sizes are roughly balanced and proportional to the test distribution sizes. While in reality such conditions may not always exist, they enable us to focus on specific aspects of the problem such as how the different classifiers respond to different levels of distinction granularity. The classification experiments carried out in this study are all binary and documents are chosen such that there is no ambiguity with respect to a given distinction, i.e. documents that belong to both categories in a particular classification task are excluded.

Our dataset is comprised of 4 coarse, 8 medium and 20 fine-grained categories, a total of 8,000 documents in the coarse-grained categories, 16,000 in the medium and 40,000 in the fine-grained categories. There are 6 coarse, 8 medium and 17 fine-grained separate binary classification tasks. The category descriptions are listed in Table 3.1 while each set of binary classification tasks is shown in Table 3.2.

The following pre-processing steps were carried out on all the data: 1) *tokenise*, 2) *remove list enumerators and markers*, 3) *remove punctuation*, 4) *remove numerical values*

	Cat Code	Description
Fine	C151	ACCOUNTS/EARNINGS
	C152	ANNUAL RESULTS
	C171	SHARE CAPITAL
	C172	BONDS/DEBT ISSUES
	C173	LOANS/CREDITS
	C174	CREDIT RATINGS
	C181	MERGERS/ACQUISITIONS
	C182	ASSET TRANSFERS
	C183	PRIVATISATIONS
	C311	DOMESTIC MARKETS
	C312	EXTERNAL MARKETS
	E211	EXPENDITURE/REVENUE
	E212	GOVERNMENT BORROWING
	E511	BALANCE OF PAYMENTS
	E512	MERCHANDISE TRADE
	M131	INTERBANK MARKETS
	M132	FOREX MARKETS
	M141	SOFT COMMODITIES
	M142	METALS TRADING
	M143	ENERGY MARKETS
Medium	C15	ACCOUNTS/EARNINGS
	C17	SHARE CAPITAL
	C18	OWNERSHIP CHANGES
	C31	MARKETS/MARKETING
	E21	GOVERNMENT FINANCE
	E51	TRADE/RESERVES
	M13	MONEY MARKETS
	M14	COMMODITY MARKETS
Coarse	CCAT	CORPORATE/INDUSTRIAL
	ECAT	ECONOMICS
	MCAT	MARKETS
	GCAT	GOVERNMENT/SOCIAL

Table 3.1: Category Descriptions

and 5) *remove stopwords.* We use the Cross Language Evaluation Forum (CLEF) English stopword list.[1]

3.2.2 Evaluation Measures

For this study we use standard TC evaluation measures for binary classification tasks, *accuracy* and *recall*:

$$accuracy = \frac{\text{TP}}{\text{T}} \qquad recall(c) = \frac{\text{TP for class } c}{\text{T for class } c}$$

where TP is the number of true positives and T the total number of documents. When an overall picture of a classifier's performance is required we will usually just report accu-

[1] http://www.unine.ch/info/clef

Fine	**Medium**	**Coarse**
C151 / C152	C15 / C17	CCAT / ECAT
C171 / C172	C15 / C18	CCAT / MCAT
C171 / C173	C15 / C31	CCAT / GCAT
C171 / C174	C17 / C18	ECAT / MCAT
C172 / C173	C17 / C31	ECAT / GCAT
C172 / C174	C18 / C31	MCAT / GCAT
C173 / C174	E21 / E51	
C181 / C182	M13 / M14	
C181 / C183		
C182 / C183		
C311 / C312		
E211 / E212		
E511 / E512		
M131 / M132		
M141 / M142		
M141 / M143		
M142 / M143		

Table 3.2: Binary Classifications

racy, while in cases requiring more detailed analysis we will also consider the individual category recall scores. To assess a classifier's performance over a complete set of M binary classifications, we use *macro-averaged accuracy*:

$$MAA = \frac{1}{M} \sum_{j=1}^{M} accuracy_j$$

Note that *macro*-averaged accuracy is equivalent to *micro*-averaged accuracy in the case that all classes have an equal number of test samples, as in our experiments.

3.2.3 Classifiers

We investigate three classification techniques in our experiments:

- *Support Vector Machines* (SVMs) are arguably the most successful discriminative classification technique to date for the TC task.
- *Boosting* is a discriminative classification technique that has gained popularity in recent years and claims state-of-the-art performance on a number of TC experimental corpora.
- *Multinomial Naïve Bayes* (MNB) is the most commonly employed generative classification technique with certain attractive properties, though its performance on the TC task usually trails its discriminative counterparts.

There has been some recent interest in finding ways to improve the performance of the MNB classifier, whilst maintaining its simplicity and efficiency (Kim, Rim, Yook & Lim 2002, Rennie et al. 2003). We use our own MNB implementation, and propose techniques for improving the document representation (beyond traditional feature selection),

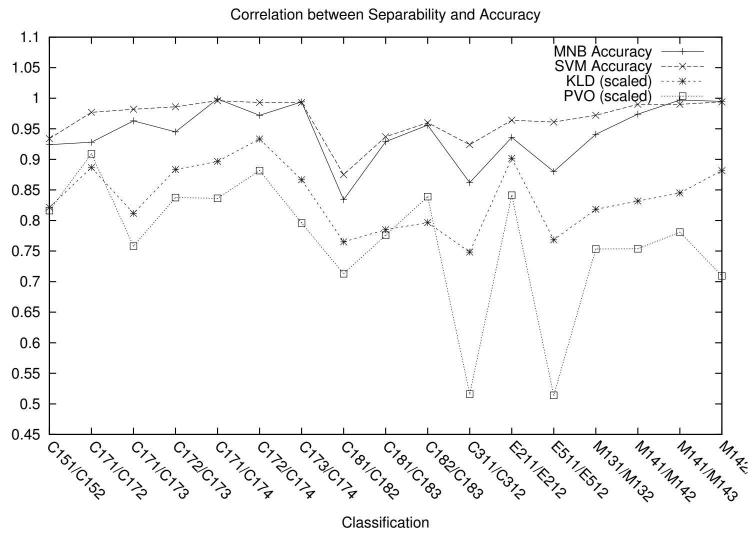


Figure 3.1: Sep/Acc correlation over fine grained categories.

demonstrating a significant performance improvement without any modifications to the algorithm itself.

We use SVM^{light} (1.3) with the radial base function (RBF) kernel (default width) and scaled, unnormalised document vectors. This configuration slightly outperformed the popular combination of the linear kernel with tfc-weighted input vectors (eg. (Yang & Liu 1999)). We also found that the default value of the regularization parameter C performed well in comparison to other choices for both the linear and RBF kernels. All other parameters are set to their default values.

We use *BoosTexter* (Schapire & Singer 2000), an implementation of the *AdaBoost* algorithm for text classification. We use the best-performing *real AdaBoost.MH* variant and as in (Schapire & Singer 2000) allow 10,000 rounds of boosting for each classification task. Feature values are per document term frequency counts. We implement our own version of the MNB classifier using balanced class priors.

3.3 Investigating Granularity

We use two metrics for estimating category separability. The first measures the quantity of overlap between two categories in terms of vocabulary usage and is called *percent vocabulary overlap* (PVO). The PVO between two categories C_1 and C_2, where $\mathbf{T_i}$ is the set of all terms occuring in the training data for category C_i, is defined as:

$$\mathrm{PVO}(C_1, C_2) = \frac{|\mathbf{T_1} \cap \mathbf{T_2}|}{|\mathbf{T_1} \cup \mathbf{T_2}|} * 100$$

Advantageously, PVO is self-normalising and intuitively easy to interpret. Disadvantageously, it doesn't take word frequency into account, considering only the presence or absence of a term in a distribution. The second measure we use is *Kullback-Liebler divergence* ($\mathrm{D_{KL}}$), a widely-used method of estimating the 'distance' between two probability distributions. Given distributions P and Q, the Kullback-Liebler divergence of Q from P

Table 3.3: Investigating Granularity

	Categories	SVM Accuracy	$\mathbf{D_{KL}^{AVG}}$	PVO
Fine	C151 C152	0.934	0.73	28.28
	C171 C172	0.977	1.12	25.31
	C171 C173	0.981	0.67	30.14
	C172 C173	0.987	1.10	27.6
	C171 C174	0.996	1.18	27.64
	C172 C174	0.993	1.40	26.19
	C173 C174	0.994	1.00	28.93
	C181 C182	0.876	0.39	31.59
	C181 C183	0.942	0.51	29.57
	C182 C183	0.957	0.58	27.55
	C311 C312	0.925	0.29	37.89
	E211 E212	0.966	1.21	27.48
	E511 E512	0.963	0.41	37.95
	M131 M132	0.975	0.71	30.29
	M141 M142	0.990	0.79	30.28
	M141 M143	0.992	0.87	29.41
	M142 M143	0.994	1.09	31.7
	Average	**0.967**	**0.83**	**29.9**
Med	C15 C17	0.977	1.02	26.84
	C15 C18	0.972	0.77	27.25
	C15 C31	0.967	0.99	25.33
	C17 C18	0.967	0.75	26.94
	C17 C31	0.980	1.15	23.46
	C18 C31	0.967	0.75	28.22
	E21 E51	0.973	0.77	31.65
	M13 M14	0.993	0.96	27.32
	Average	**0.974**	**0.90**	**27.1**
Coarse	CCAT ECAT	0.978	0.88	26.90
	CCAT MCAT	0.990	0.97	24.09
	CCAT GCAT	0.988	1.22	22.08
	MCAT ECAT	0.979	1.02	27.01
	MCAT GCAT	0.992	1.30	22.03
	ECAT GCAT	0.974	1.05	26.57
	Average	**0.983**	**1.07**	**24.8**

is defined as:

$$\mathrm{D}_{\mathrm{KL}}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

In our case we are interested in the divergence between the class conditional distributions of the two categories in question. We use add-1 (Laplacian) smoothing to handle unseen events and take the 1,000 highest-probability features from each category. D_{KL} is asymmetric, i.e. $\mathrm{D}_{\mathrm{KL}}(P||Q) \neq \mathrm{D}_{\mathrm{KL}}(Q||P)$ so we measure the average (or symmetric) D_{KL} between the two categories:

$$\mathrm{D}_{\mathrm{KL}}^{\mathrm{AVG}}(P, Q) = \frac{1}{2}(\mathrm{D}_{\mathrm{KL}}(P||Q) + \mathrm{D}_{\mathrm{KL}}(Q||P))$$

Table 3.4: Correlation Metrics

	Correlation Coef.		**Spearman's Rank**	
	MNB	**SVM**	**MNB**	**SVM**
D_{KL}^{AVG}	0.6775	0.7057	0.5711	0.7304
PVO	0.5404	0.3752	0.2279*	0.2451*

Table 3.5: P(*term*|*class*) ranking for C172/4 terms

Rank	**C172**	**C174**
1	PAY	Sale
2	DATE	Announcement
3	ISS	CARE
4	TYPE	Date
5	AMT	Rating
6	BORROWER	downgrade
7	NOTES	Aa
8	BP	Amount
9	MLN	rev
10	FEES	assigns
11	LISTING	implications
12	FULL	FD
13	DENOMS	lowered
14	FREQ	DBRS
15	SPREAD	Issuer
16	SALE	affirmed
17	GOV	CBRS
18	LAW	AM
19	CRS	Expected
20	NEG	TX

Our use of D_{KL} in this context is somewhat similar to (Kilgarriff & Rose 1998) who use it to measure corpus similarity. There is also related work in the IR field in the area of quantification of query performance (He & Ounis 2006, Cronen-Townsend, Zhou & Croft 2002), query difficulty (Amati, Carpineto & Romano 2004, Yom-Tov, Fine, Carmel & Darlow 2005) and query ambiguity (Cronen-Townsend & Croft 2002). In each of these studies the basic paradigm is the same; a distributional similarity metric is used to compare queries with document collection language models. Positive correlation is demonstrated between query performance and distance from the document collection distribution as measured by the chosen metric (e.g. relative entropy). There is an evident synergy between this work and ours in terms of the applying information theoretical techniques to measure relationships between distributions as a correlate for intrinsic model performance, and it is helpful to observe the effectiveness of similar principles in different domains.

Table 3.3 lists the results of applying the category separability measures to each of the classification distinctions in our dataset, along with the accuracy of the SVM classifier. The correlation between PVO, D_{KL}^{AVG} and Accuracy is highlighted by Figure 3.1. Both separability measures are scaled (and PVO is inverted) to fall within the same range

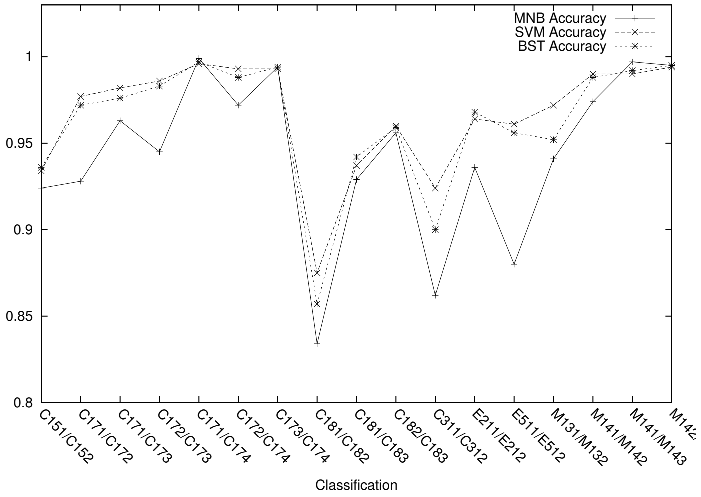


Figure 3.2: Fine grained classifier accuracy

as Accuracy. The correlation is quantified in Table 3.4 using the standard correlation coefficient, as well as Spearman's rank correlation coefficient. The asterisk denotes not significant at $p < 0.05$ and both metrics range between -1, no correlation and 1, perfect correlation. As expected, classification accuracy is more closely correlated with $\mathrm{D}_{\mathrm{KL}}^{\mathrm{AVG}}$, due to its account of term frequency, than with PVO. Interestingly, there is a closer correlation between $\mathrm{D}_{\mathrm{KL}}^{\mathrm{AVG}}$ and SVM accuracy than between $\mathrm{D}_{\mathrm{KL}}^{\mathrm{AVG}}$ and MNB accuracy, which is somewhat surprising bearing in mind that from a theoretical perspective $\mathrm{D}_{\mathrm{KL}}^{\mathrm{AVG}}$ is more closely related to MNB.

From Table 3.3 it can be seen that when averaging over the classifications at a given level of granularity, the separability of the categories increases as the granularity gets coarser, correlating with the accuracy of the classifier. Within each level of granularity, however, there is significant deviation. For instance, while the fine-grained level contains the narrowest distinctions and hardest classification tasks, it also contains the most widely-separated distinction according to the D_{KL} measure, represented by the categories C172 (BONDS/DEBT ISSUES) and C174 (CREDIT RATINGS). It is informative to look at the terms from each of these distributions with the highest class conditional likelihood rankings (Table 3.5).

It turns out that though these categories share a similar topic domain, the use of capitalisation in C172 helps to distinguish it from C174. This is a specific example of a general principle in TC, which is that even when two categories are conceptually similar, they may exhibit specific differences that are represented by a small but highly discriminative subset of the vocabulary. Affirming this perspective is recent work by Bouma & de Rijke (2006) in which experimental analysis leads to the conclusion that classifying documents into narrower categories can yield better results than broad categories due to the association of narrow categories with highly specific discriminative features.

Table 3.6: Accuracy (MAA) Comparison

Granularity	Category	MNB	SVM	BST
Fine	C151 C152	0.924	0.934	0.936
	C171 C172	0.928	0.977	0.972
	C171 C173	0.963	0.981	0.976
	C172 C173	0.945	0.987	0.983
	C171 C174	0.999	0.996	0.997
	C172 C174	0.972	0.993	0.988
	C173 C174	0.994	0.994	0.994
	C181 C182	0.834	0.876	0.857
	C181 C183	0.929	0.942	0.942
	C182 C183	0.956	0.957	0.959
	C311 C312	0.862	0.925	0.900
	E211 E212	0.936	0.966	0.968
	E511 E512	0.880	0.963	0.956
	M131 M132	0.941	0.975	0.952
	M141 M142	0.974	0.990	0.988
	M141 M143	0.997	0.992	0.992
	M142 M143	0.995	0.994	0.995
	MAA	**0.943**	**0.967**	**0.962**
Med	C15 C17	0.973	0.977	0.972
	C15 C18	0.962	0.972	0.970
	C15 C31	0.970	0.967	0.968
	C17 C18	0.940	0.967	0.955
	C17 C31	0.984	0.980	0.982
	C18 C31	0.965	0.967	0.968
	E21 E51	0.958	0.973	0.968
	M13 M14	0.995	0.993	0.987
	MAA	**0.968**	**0.974**	**0.971**
Coarse	CCAT ECAT	0.980	0.978	0.964
	CCAT MCAT	0.990	0.990	0.983
	CCAT GCAT	0.991	0.988	0.991
	MCAT ECAT	0.980	0.979	0.977
	MCAT GCAT	0.993	0.992	0.986
	ECAT GCAT	0.959	0.974	0.980
	MAA	**0.982**	**0.983**	**0.980**

3.4 Classifier Comparison

We now compare the performance of the three classification techniques. It is known that the performance of MNB often degrades in the presence of imbalanced numbers of training samples (Rennie et al. 2003). However, this is not an issue in our experiments as we have deliberately selected an equal number of training samples for each class.

Table 3.6 compares classifier accuracy averaged over each level of granularity. In the coarse and medium-grained cases, there is no significant difference between the classifiers (according to an S- and T-test (Yang & Liu 1999)). In the fine-grained case, both SVM and BST significantly outperform MNB according to an S-test ($p < 0.05$) and SVM

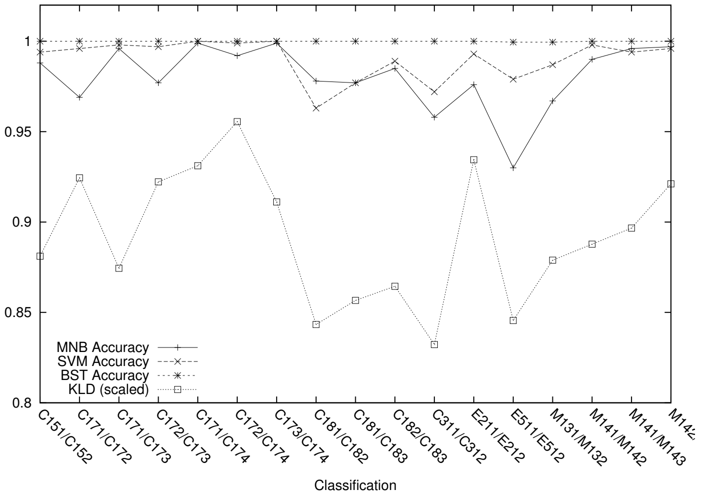


Figure 3.3: Training accuracy

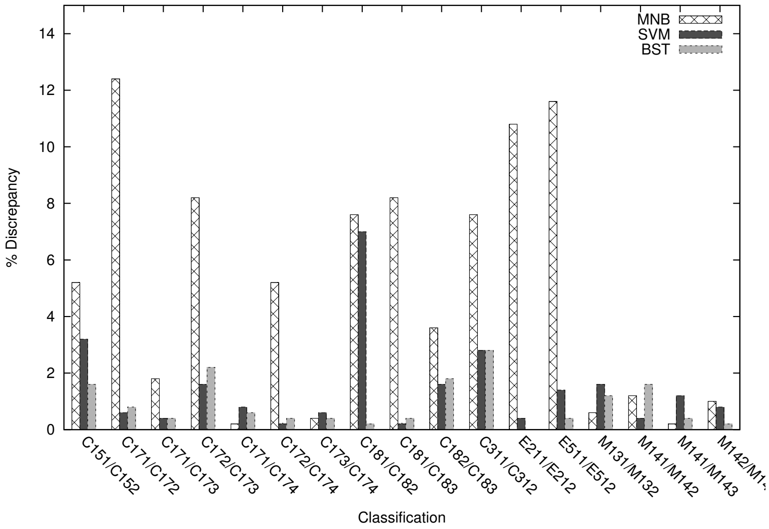


Figure 3.4: Recall balance for fine grained classes

significantly outperforms MNB according to a T-test ($p < 0.1$). There is no significant difference between SVM and BST.

Figure 3.2 illustrates the performance of the classifiers over each fine-grained task. The accuracy of MNB fluctuates more than the other techniques and is in general inferior except in cases where the separability of the categories is high. The superiority of SVM over BST is only really apparent in cases where the category separability is lowest. Figure 3.3 plots the training accuracy of the classifiers on the fine-grained data. It is interesting to note that in the more difficult fine-grained instances SVM training error increases, allowing a wider margin and consequent variance control, whereas BST always converges to near perfect training accuracy.

Previous research has shown that discriminative techniques such as SVM and BST outperform MNB in whole-corpus TC experiments, but the low performance of MNB in such studies is often exacerbated by highly skewed training sample sizes, an issue

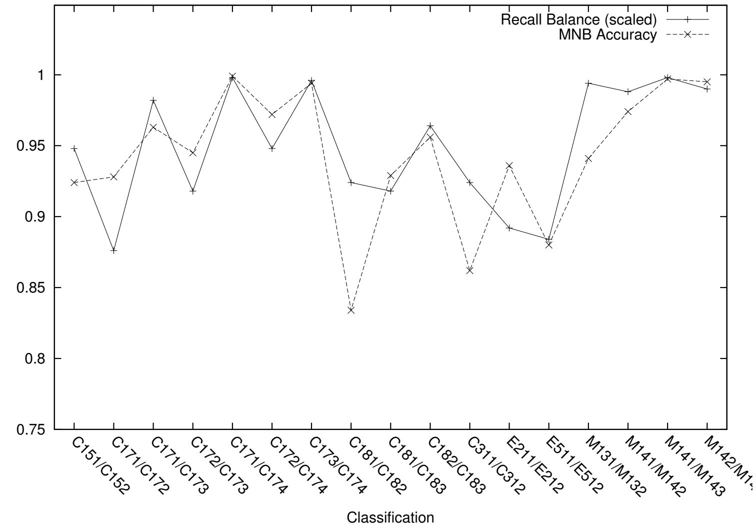


Figure 3.5: MNB accuracy and recall balance

addressed by (Rennie et al. 2003). Our results suggest that given an adequate, well-balanced training set, strong discriminative classifiers still yield better results than MNB, but only significantly so when the categories are harder to separate. In the medium and coarse-grained instances MNB performs competitively. This suggests that a simple measure of category separability, such as those presented in this study, could be a guide as to the type of classifier best suited to the problem.

Figure 3.4 displays the balance achieved by the classifiers between the two categories (in terms of recall) in each classification instance. It can be seen that the SVM and BST achieve a consistent balance (low % discrepancy) in most cases (C181/C182 is somewhat anomalous), while MNB fluctuates quite significantly. These observations follow from the theoretical properties of the models in that discriminative optimisation explicitly seeks a balance between the classes by finding a separating hyperplane with low empirical error, whereas MNB relies on the integrity of the training distributions to achieve this balance.

Figure 3.5 highlights the rough correlation between recall balance and accuracy for MNB, suggesting that a major reason for MNB's weak performance is that even with seemingly well-balanced training data it still exhibits a tendency to disproportionately prefer one category over another, an observation consistent with Rennie et al. (2003).

3.5 Document Representation

We are interested in investigating the impact of improving the underlying document representation (DR henceforth) on classification performance. Our strategy is threefold: 1) select salient document regions 2) balance the training distributions and 3) introduce linguistically motivated higher-order features. We will focus on performance over the fine-grained classification tasks where the error margin leaves more room for visible improvement, and within which there is a wide range of category separability.

Table 3.7: Comparing Feature Selection Techniques with SVM

FS Scheme:	None	Stop	$\chi^2_{max}(3000)$	P/N/V/A	P/N/V	P/N
SVM MAA	0.953	0.967	0.965	0.965	0.965	0.962
Approx Reduction:	0%	60%	80%	55%	60%	65%

Stop = stop word removal
P/N/V/A = retain proper names, nominals, lexical verbs and adjectives

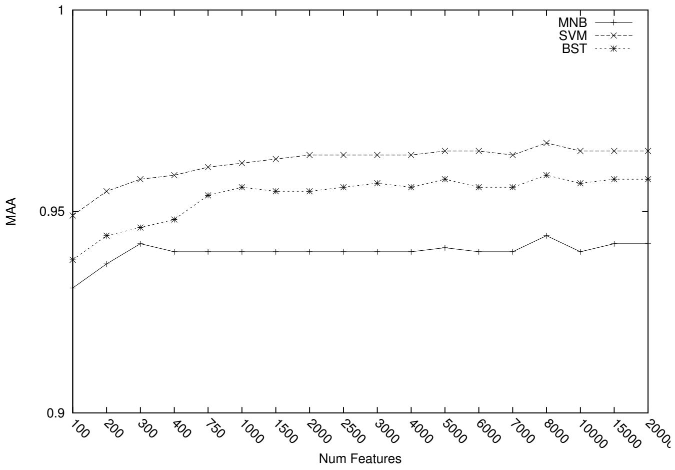


Figure 3.6: Feature Selection with χ^2_{max}

3.5.1 Feature Selection

Feature selection (FS) is the traditional method for reducing the amount of noise in a distribution and protecting classifiers from overfitting the training data. Techniques for FS have been extensively studied in previous work, eg. (Mladenic & Grobelnik 1999, Yang & Pedersen 1997). Figure 3.6 plots macro-averaged accuracy (MAA) on the fine-grained development data as a function of the global number of features chosen at various thresholds using the χ^2_{max} feature selection metric (Yang & Pedersen 1997). The discriminative classifiers follow a similar pattern, displaying a gradual increase in performance as the number of features increases, leveling out at around 2000-3000, while MNB reaches an optimum at around 300 features. In these experiments, traditional FS does not significantly improve accuracy for any of the classifiers, though it is able to significantly reduce the dimensionality of the input space without loss of accuracy.

We also experiment with feature selection using linguistic criteria, i.e. by selecting only words with certain part-of-speech tags to be included as features. Table 3.7 compares the accuracy of SVM under various different feature selection strategies. Part-of-speech tagging is performed using the RASP tagger (1.3), and 'P/N/V/A' denotes that (P)roper names, (N)ominals, lexical (V)erbs and (A)djectives were retained. It can be seen that in terms of accuracy and data reduction, traditional feature selection with χ^2_{max} performs better than filtering by part-of-speech. This is to be expected when we consider that term goodness metrics such as χ^2 calculate the informativeness of a term with respect to the task in hand, whereas part-of-speech selection is only able to choose generally informative terms.

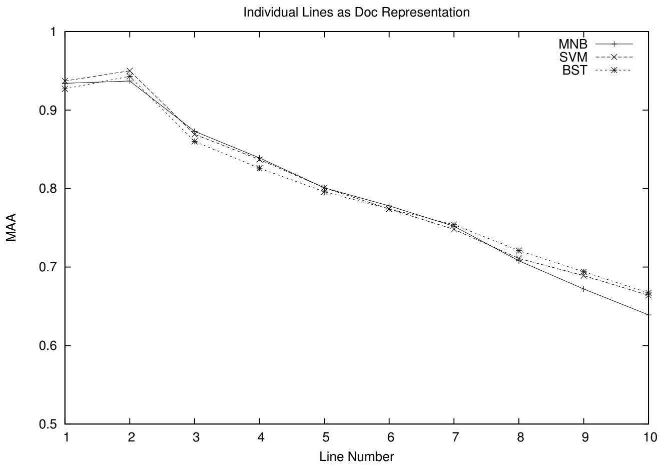


Figure 3.7: Individual line document representation

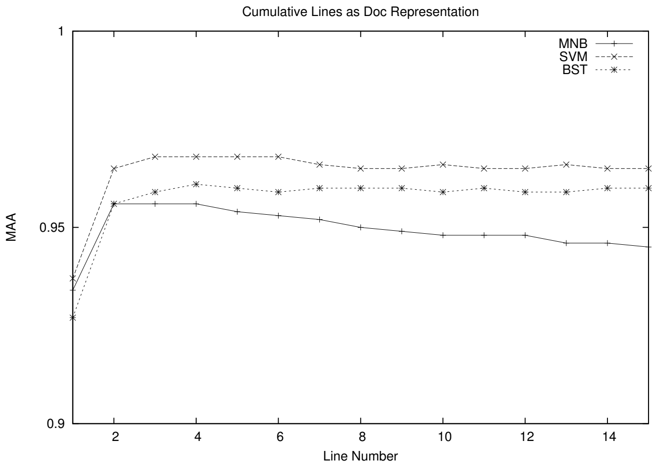


Figure 3.8: Cumulative lines document representation

3.5.2 Salient Region Selection

In the newswire domain, the topic of a story tends to be introduced within the first few lines, and the remainder of the story serves to 'fill in' the details. In general, it is usually the case that certain document regions are more salient with respect to topic than others. Figure 3.7 plots the MAA of the classifiers on the fine-grained development data when a single line is chosen to represent the document (line number shown on the x-axis). We should note that the MAA of latter lines is somewhat reduced by the fact that some documents contain fewer than ten lines. However, it is clear that the lines at the beginning of the document are significantly more informative with respect to the document category than those toward the end.

It is also informative to measure the accuracy of the classifiers as a function of *cumulative* lines. Once again we use the fine-grained development data for Figure 3.8 to plot the accuracy of the classifiers as document lines are gradually accumulated. As the more noisy latter lines are added, accuracy for the discriminative classifiers remains fairly consistent, though there is no significant improvement. Conversely, MNB reaches a peak

Table 3.8: Salient Region Selection

	MNB	SVM	BST
First 2 Lines	0.957	0.967	0.954
First 4 Lines	0.956	0.967	0.960
Whole document	0.943	0.967	0.962

after the first couple of lines, and subsequent accumulation gradually degrades its accuracy. This is an informative result for a number of reasons. Firstly it suggests that for newsire topic classification, filtering out all but the first few lines is a cheap way of significantly reducing the feature space and the amount of noise, thereby increasing classifier efficiency and accuracy. It also highlights the effectiveness of discriminative, wide margin classifiers at minimising the detrimental effects of noisy data.

Table 3.8 shows the performance of MNB and the SVM on the fine-grained test data given just the first 2 and the first 4 lines as the DR. Using just the first 2 lines represents a reduction of approximately 85-90% in total data volume, and the first 4 lines a 60-65% reduction. There is no statistically significant difference in the performance of SVM and BST when using the reduced DR; however the accuracy of MNB is significantly improved according to an S-test ($p < 0.1$). In this study we perform salient region selection across the whole dataset based on prior domain knowledge. In theory, however, it could also be carried out automatically by a technique that selected the most salient regions for each document individually through analysis of, for instance, vocabulary shift. This is an avenue for further research.

3.5.3 Balancing the Training Distributions

Even when the number of samples per training category is balanced, the actual size of the distributions may remain somewhat imbalanced as a result of varying document lengths. We hypothesise that to some extent this accounts for the excessive recall discrepancy (Figure 3.4) displayed by MNB in certain instances, and its consequent poor performance. We propose to balance the distribution sizes by selecting an equal number of features to represent each document throughout a given category such that the total number of features observed for each category is roughly the same. For example, if category A has a total training distribution size (number of observed samples) x and category B has distribution size $2 \cdot x$ then the number of features chosen for the sample representation in B will be twice that of A.

The purpose of this approach is twofold: firstly it normalises for document length variation by standardising the number of features chosen to represent samples across a given category, and secondly it normalises for variation in the quantity of training data observed for each category under the intuition that if we have observed less training data for category A than category B, we can expect category A to contain, on average, fewer relevant features per sample. In our case, because we have an equal number of training samples for each category, selecting the same number of features to represent each document across the entire collection will result in balanced training distributions, as long as the total number of features in individual documents does not fall significantly below this value.

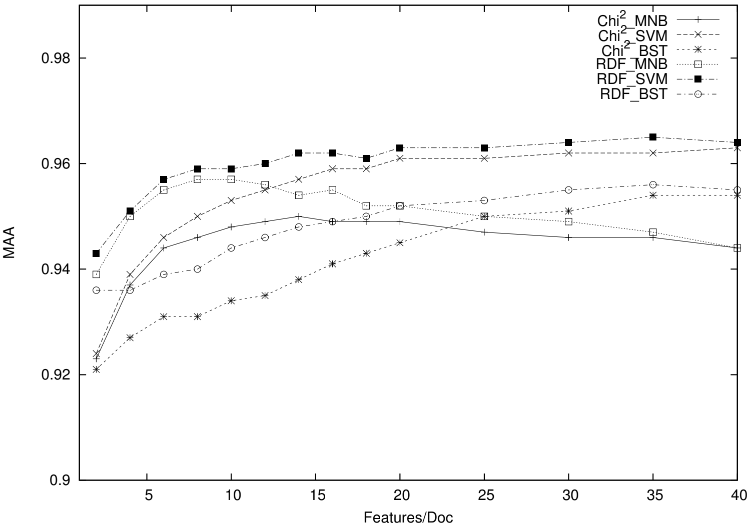


Figure 3.9: Feature Selection with Distribution Balancing

Table 3.9: Distribution Balancing vs. Traditional FS

	Distr. Balancing			**Traditional FS**		
	MNB	SVM	BST	MNB	SVM	BST
Parameter:	10	215	215	300	8000	8000
MAA	0.956	0.966	0.962	0.942	0.966	0.960

As in traditional feature selection, we want to choose informative features to represent each document; specifically we will choose the n most informative features for each document where n is preselected manually or by means of a parameter optimisation scheme and features are ranked according to their informativeness with respect to the given classification task. Again we use χ^2_{max}, as well as introducing a simple new method for term selection, called *relative document frequency* (RDF), defined as follows:

Given a binary classification problem with classes C_1 and C_2 and term t, where A is the document frequency of term t in class C_1 and B the document frequency of term t in class C_2, relative document frequency is defined as:

$$\mathrm{RDF}(t) = \mathrm{abs}[\log(A + K) - \log(B + K)]$$

K is a positive constant to avoid $\log(0)$ and can also be interpreted as a smoothing factor, avoiding the problems encountered with χ^2 for low frequency counts. We use K=5 for our experiments. Disadvantageously, this formulation of RDF can only be used for binary classification problems and, because it doesn't model distribution size, it is liable to give misleading results if the distributions are highly skewed.

Figure 3.9 plots classification accuracy as a function of the number of features selected per document (n) using the χ^2_{max} and RDF metrics. The combination of RDF with distribution balancing is particularly effective at boosting the accuracy of MNB when small values of n are chosen. Table 3.9 compares results for distribution balancing against traditional feature selection. The parameter values represent the number of features chosen, per document in the distribution balancing case and overall in the traditional case, and this parameter is tuned on the development data. We observe that in comparison with

Table 3.10: Distribution Balancing + Salient Region Selection

	DB + SRS			**Original Doc Repr**		
	MNB	SVM	BST	MNB	SVM	BST
Parameter:	10	30	50			
Fine	0.962	0.970	0.961	0.943	0.967	0.962
Medium	0.974	0.977	0.974	0.968	0.974	0.971
Coarse	0.982	0.983	0.978	0.982	0.983	0.980

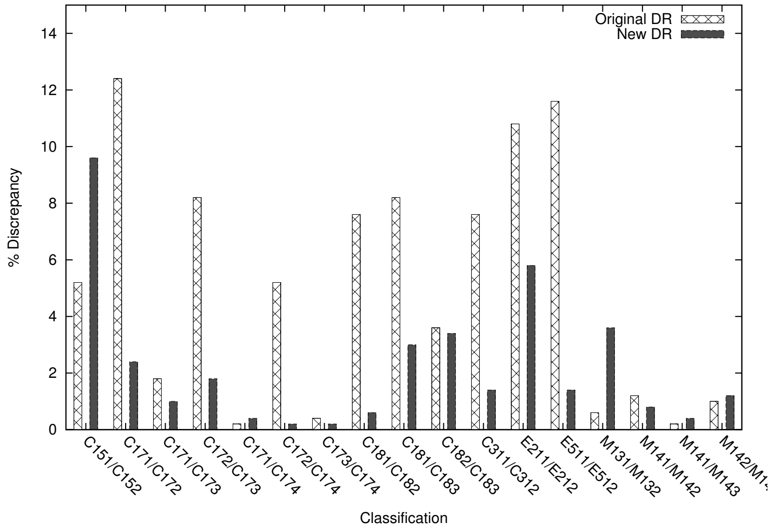


Figure 3.10: MNB Recall Balance with New DR

standard feature selection, distribution balancing has no significant effect on the accuracy of SVM and BST, which is unsurprising considering such techniques do not explicity rely on balanced training data.

Table 3.10 displays the results obtained by combining distribution balancing and salient region selection (DB & SRS henceforth) on each of the category granularities. We use the first 4 lines of each document as the salient region, and the parameter governing the number of features chosen to represent each document for the respective classification model is given in the table. As we would expect, improving the empirical distributions has the most significant effect when distinguishing less separable categories. Using the new DR, there is now no significant difference between MNB and the other classifiers according to the S- and T-tests.

Figure 3.10 shows the change in recall balance for MNB when using DB and RS. In almost every case, recall discrepancy has been reduced, quite dramatically in some cases. This correlates in general with classification accuracy and lends weight to the hypothesis that the poor performance of MNB is partially due to its tendency to inadvertantly weight one class over another.

3.5.4 Introducing More Informative Features

We now investigate a number of techniques for generating combinations of terms to include as features in the DR. Similar experiments have been carried out by a number of researchers (Raskutti, Ferrá & Kowalczyk 2001, Tan, Wang & Lee 2002, Moschitti & Basili 2004, Bekkerman & Allan 2005). Such work has focused almost exclusively on bigrams, as data sparsity, low error margins and the curse of dimensionality almost always preclude any potential gain from higher-order n-grams. Results from these studies are often inconclusive, with some researchers showing significant improvements for certain classification techniques (Raskutti et al. 2001, Tan et al. 2002) though in these instances the baseline tends to be low relative to the state-of-the-art, while other studies have shown statistically insignificant improvements over state-of-the-art baselines (Bekkerman & Allan 2005).

Bearing in mind the increased dimensionality in the space of complex features, it is perhaps unsurprising that little improvement has been observed when using bigrams for topic classification on current, relatively small topic classification benchmark corpora such as the Reuters-21578 or 20 Newsgroups corpora. The space of single-term features is of dimensionality $O(V)$, in the vocabulary size, whereas for n-grams it is $O(V^n)$. Exacerbating the problem is the fact that fine-grained categories, where we might expect the additional precision offered by complex features to be of benefit, tend to be the least well represented in topic classification corpora. To our knowledge this is the first study into the application of complex features using a topic classification dataset containing reasonably well represented fine-grained category distinctions.

A common theme in previous research into the use of higher-order features is that positive results are only obtained by augmenting existing single-term representations (Tan et al. 2002, Moschitti & Basili 2004, Bekkerman & Allan 2005), rather than replacing them. We follow this principle in our experiments.

We use the distribution balancing method from section 3.5.3, selecting an equal number of representative bigrams for each document, with which we augment the best-performing single-term representations for each classifier. We investigate three techniques for generating candidate bigrams:

- *S1*: Include naturally occurring contiguous bigrams.
- *S2*: Include contiguous and non-contiguous bigrams generated from a set of highly informative single terms.
- *S3*: Include terms in binary grammatical relations as identified by RASP.

The first approach is very similar to that taken in most previous studies, where contiguous bigrams are selected from the original (stopword filtered) text based on their informativeness. For example, given the sentence *"Barclays Bank announced new job losses yesterday"*, the following candidate bigrams would be generated:

Barclays_Bank *Bank_announced*
announced_new *new_job*
job_losses *losses_yesterday*

In the second approach, we begin with an already filtered set of high-ranking single terms (for a given document) and generate all ordered bigrams within a certain window, contiguous and non-contiguous. For example, given the following collection of single terms: *imports Russian grain farming* and a window length of 4, the following bigrams would be generated:

imports_Russian *imports_grain* *imports_farming*
Russian_grain *Russian_farming* *grain_farming*

Table 3.11: Results for bigram inclusive document representations

	S1		**S2**		**S3**		**No Bigrams**	
	MNB	SVM	MNB	SVM	MNB	SVM	MNB	SVM
Fine	0.965	0.971	0.965	0.971	0.964	0.969	0.962	0.970
Medium	0.974	0.977	0.975	0.978	0.973	0.971	0.974	0.977
Coarse	0.985	0.987	0.985	0.987	0.984	0.983	0.982	0.983

Finally, in the third approach we use RASP (1.3) to generate bigrams representing pairs of terms that are grammatically related.[2] After experimentation we select the following set of grammatical relations (GRs) for inclusion in the candidate bigram set:

subject → verb
verb ← direct object
modifier → noun

The direction of the arrow denotes head←dependency, and is implicitly respected by the ordering of the terms in each bigram. For example, given the sentence *University students hold silent vigil*, the following bigrams would be generated:

University_students	(modifier → noun)
students_hold	(subject → verb)
hold_vigil	(verb ← direct object)
silent_vigil	(modifier → noun)

As far as we are aware, this method of generating candidate bigrams for text classification has not been previously investigated. Theoretically, bigrams representing terms that exist in central grammatical relationships should be good candidates, as they usually represent the main themes of the text. For instance, consider the following sentence:

> *The government of South Korea recently borrowed two million dollars to fund urban regeneration.*

If such a sentence exists within the class of documents whose topic is 'government borrowing', clearly the most informative bigram is *government_borrowed* which represents the subject ↔ verb grammatical relation. We would expect this bigram to be identified by the syntactic parser, whereas it would not be included as a candidate in the first approach (S1) and though it would occur as a candidate in the second approach (S2), a number of potentially noisy bigrams would also be included.

Table 3.11 displays the results of using the three bigram augmentation strategies (denoted by S1, S2 and S3 respectively) on the test data. All three techniques require a parameter representing the number of bigrams to include in each DR. For efficiency purposes, we tie this to the parameter for the number of single terms in the DR (30 for SVM, 10 for MNB). For S2 we use a 10 term window length to generate candidate bigrams.

The use of bigrams does improve overall performance in most cases, though only very marginally. The relatively sophisticated GR selection strategy (S3) does not yield superior results compared to the other simpler schemes. An explanation for this can be drawn from a detailed analysis of the complex features generated for a specific fine-grained classification instance:

[2] Note that the original data, without preprocessing, was used to generate the grammatical relations.

Table 3.12: Top-ranking bigrams for category C311

	S1	**S2**	**S3**
1	import_duty	imports_imports	import_Japan
2	Major_suppliers	imports_import	duty_import
3	Tokyo_Commod's	imports_Ministry	import_oil
4	data_released	import_import	import_sugar
5	oil_imports	imports_imported	see_import
6	imports_tonnes	imports_Finance	import_China
7	earlier_Finance	imports_earlier	import_wheat
8	import_duties	import_imported	total_import
9	imports_months	imports_yr/yr	quota_import
10	Ministry_data	Finance_Japan	rise_import
11	Total_Major	imports_released	fall_import
12	imports_yr/yr	imports_tonnes	import_tonne
13	sugar_imports	imports_data	raise_duty
14	Finance_Ministry	Ministry_Total	import_gold
15	import_quota	imports_showed	ban_import
16	China_imported	Finance_Total	import_crude
17	DURUM_WHEAT	Japan_Total	duty_custom
18	Commodities_Desk	Finance_earlier	import_coffee
19	yr/yr_Japan	imports_Japan	import_total
20	Cumulative_imports	imports_duty	wheat_quality
21	crude_imports	imports_tariff	import_corn
22	gold_imports	Ministry_data	import_product
23	wheat_imports	import_duties	cut_import
24	imports_rise	imports_customs	import_Jan
25	Warsaw_Newsroom	import_duty	duty_wheat

Table 3.12 displays the top-ranking bigrams (under the RDF metric) for the category C311 – *Domestic Markets* (versus C312 – *External Markets*), using the three candidate selection strategies. S1 generates the types of bigrams we would expect, while S2 yields more obscure bigrams, linking seemingly redundant alternative morphological forms of the same word (*imports_import*) along with terms that don't appear to be (directly) semantically related in the text (*imports_Finance*, *Finance_Japan*). Hence, rather than creating features that represent finer grained conceptual information, the S2 features are mostly co-ocurrences of possibly unrelated discriminative single terms. Despite this, it performs equally as well as either of the other techniques.

On the other hand, the GR selection scheme, S3, captures seemingly important domain concepts represented by terms occuring in central grammatical relationships, and yet performs no better than the other strategies. This is understandable when the nature of the category distinction is reconsidered. We have suggested that relatively fine-grained categories are harder to distinguish than coarse-grained; however, the distinctions between fine-grained categories are often not actually particularly 'fine-grained'. For example, documents in the C311 category generally contain information about imports, and the *type* of import is of no significance. This renders many of the top-ranking S3 bigrams redundant (*import_oil, import_sugar, import_wheat...*) because the single term *import* supersumes all of them with greater generality.

The fact that none of the bigram generation strategies outperforms the single term representation by more than a marginal amount suggests that none of the distinctions in the RCV1 corpus (and probably in newswire topic categorization in general) are semantically fine-grained in nature. Seemingly significant improvements in topic categorization performance for 'weaker' classification techniques (such as MNB) through the inclusion of higher-order features in the DR may not actually represent an improvment in the informativeness of the features; rather it may simply be the case that the new DR results in more balanced empirical distributions and hence a more reliable result from the BDR. Similar improvement may well be obtainable through simple distribution improvement techniques such as the ones we have presented in this study.

Our experiments suggest that term-combination features such as bigrams will not significantly enhance TC accuracy in the newswire domain with a strong classifier and/or relatively noise-free document representation. However, this does not preclude the possibility that TC instances in other domains may genuinely benefit from the inclusion of complex features.

3.6 Complexity

The distribution balancing technique we have presented is $O(nk^2)$ in the number of training samples n and an upper bound k on the number of features per sample. This is derived from the calculation of the term-goodness metric, which for χ^2 and RDF is $O(Vm)$ in the number of classes m and the vocabulary size V, and the feature ranking for each document representation, $O(nk^2)$), given the reasonable assumption that $n \propto Vm$. Using quicksort yields an average complexity of $\Theta(k \log k)$ for the feature ranking, and because k is independent of n and usually relatively small, we can consider it a constant of the algorithm, yielding a complexity for DB of $O(n)$. Salient region selection is computationally trivial in our formulation, and thus DB+SRS in combination with MNB ($O(n)$) results in an overall time complexity of $O(n)$, which easily scales to training on large datasets, such as the whole of RCV1 ($\sim$800,000 documents), which would stretch the bounds of feasibility for more complex optimization procedures such as the SVM.

The complexity of both the S1 and S2 bigram candidate selection schemes is order linear, though S2 can become quite costly as it generates $\Sigma_{i=1}^{n-1} \min(n-i, w)$ candidate bigrams for an input set of n single terms and a window length of w. S3 incurs additional costs due to the parser, and yielded a similar number of candidate bigrams to S2 in our experiments.

3.7 Conclusions and Future Work

We have demonstrated through empirical analysis the correlation between category separability and classification accuracy, and suggested that an efficient calculation of separability could be used to guide classifier selection.

On the one hand, our results confirm the superiority of wide-margin discriminative classification models for TC, in that SVM and BST generally outperform MNB, even without the usual issues of highly-skewed training data. On the other hand, we have shown that with some simple distribution enhancement techniques MNB can be competitive with the more complex techniques, whilst retaining its advantages in terms of efficiency and perspicuity.

Our hypothesis about the potential benefit of using complex features for fine-grained topic classification proved to be ill-founded. Analysis demonstrated that this was largely due to a discrepancy between our intuitions about 'fine-grainedness' and the actual semantic distinctions between so called fine-grained newswire categories. As a result, our experiments have shown that even with well represented fine-grained categories, little improvement is gained through the inclusion of bigram features.

We hope to see more work carried out on investigating how particular techniques for topic classification scale to very large datasets, both from a theoretical and practical 'real world' perspective, eg. (Yang, Zhang & Kisiel 2003, Joachims 2006). It would also be interesting to see what effect our techniques for salient region selection, distribution balancing and complex feature generation have on more advanced generative classification models.

Chapter 4

Spam Filtering

In this chapter, we consider the topical problem of automatic spam filtering. Motivated by current efforts to construct more realistic spam filtering experimental corpora, we present a newly assembled, publicly available corpus of genuine and unsolicited (spam) email, dubbed *GenSpam*. We also propose an adaptive classification model for semi-structured documents based on language model component interpolation. We compare this with a number of alternative classification models, and report promising results on the spam filtering task using a specifically assembled test set to be released as part of the *GenSpam* corpus. The work presented in this chapter is also published in the proceedings of the third conference on email and anti-spam – *CEAS 2006* (Medlock 2006*a*).

4.1 Introduction

The well-documented problem of unsolicited email, or *spam*, is currently of serious and escalating concern.[1] In lieu of effective legislation curbing the dissemination of mass unsolicited email, *spam filtering*, either at the server or client level, is a popular method for addressing the problem, at least in the short-term. While various spam filters have begun to find their way onto the market, there is a lack of rigorous evaluation of their relative effectiveness in realistic settings. As a result, there is an ongoing research effort to construct representative, heterogeneous experimental corpora for the spam filtering task. In this paper, we present a sizeable, heterogeneous corpus of personal email data to add to the spam filtering research arsenal, dubbed *GenSpam*.[2] We also present and evaluate an adaptive LM-based classification model for spam filtering, or more generally semi-structured document classification.

4.2 Related Work

Some of the first published work on statistical spam filtering was carried out by (Sahami et al. 1998) using a multi-variate Bernoulli NB model. However, the training and test sets were small (less than 2000 total messages), and not publicly available, thus rendering the experiments non-replicable.

[1] See research by *MessageLabs* (http://www.messagelabs.co.uk) and *Ferris* (http://www.ferris.com).

[2] Available from http://www.cl.cam.ac.uk/users/bwm23/

Androutsopoulos et al. (2000) present results for spam filtering on the *LingSpam* corpus. They compare a multinomial NB classifier with a kNN variant, the results favouring NB. Carreras & Marquez (2001) build on this work, publishing improved results on the same corpus using boosting decision trees with the *AdaBoost* algorithm.

Drucker et al. (1999) publish results comparing the use of SVM's with various other discriminative classification techniques on the spam filtering problem, with binary-featured SVM's and boosting decision trees performing best overall. Unfortunately the test sets they used are not publicly available.

The *LingSpam* corpus (Androutsopoulos et al. 2000) is currently the most widely-used spam filtering dataset. It consists of messages drawn from a linguistics newsgroup, and as such the genuine messages are largely homogeneous in nature (linguistic discussion) and thus non-representative of the general spam-filtering problem, where genuine messages typically represent a wide range of topics. Additionally, the corpus consists predominantly of genuine messages (2412 genuine, 481 spam) whereas in reality the balance is more often in favour of spam, and is too small to allow experimentation into the important issue of how a classifier *adapts* as the nature of spam and/or genuine email changes over time and between different users.

In light of the inadequacy of *LingSpam* and the paucity of publicly available, realistic email data for experimental spam filtering, various efforts have recently been made to construct more realistic spam filtering experimental corpora, most notably the TREC 2005 spam track corpus and a variant of the Enron corpus (Klimt & Yang 2004, Cormack & Lynam 2005). Such efforts have provided opportunity for a new generation of more realistic spam filtering experiments.

The spam filtering problem has traditionally been presented as an instance of a *text categorization* problem on the basis that most email contains some form of identifiable textual content. In reality, the structure of email is richer than that of flat text, with meta-level features such as the fields found in MIME compliant messages. Researchers have recently acknowledged this, setting the problem in a *semi-structured document classification* framework. Bratko & Filipič (2004) take this approach on the *LingSpam* corpus, reporting a significant reduction in error rate compared with the flat text baseline.

The semi-structured document classification framework is, of course, applicable to a wider range of problems than just spam filtering, as in (Yi & Sundaresan 2000, Denoyer & Gallinari 2004, Bratko & Filipič 2004). In all these cases the NB classification model is extended to take account of the componential document structure in question. We note that the limiting *conditional independence assumption* of NB can be relaxed in a classification framework based on smoothed higher-order n-gram language models. This is also recognised by Peng & Schuurmans (2003), who report state-of-the-art results using a higher-order n-gram based LM text classifier on a number of data sets. We define a similar classification model, but extend it into an adaptive semi-structured framework by incorporating recursive structural component *interpolation*. We apply the resulting classification model to the newly assembled *GenSpam* email corpus.

4.3 A New Email Corpus

The corpus we have assembled consists of:

- 9072 genuine messages (∼154k tokens)
- 32332 spam messages (∼281k tokens)

The imbalance in the number of messages is due in part to the difficulty of obtaining genuine email - persuading people to donate personal email data is a challenge. On the whole though, spam messages tend to be significantly shorter than genuine ones, so in terms of total content volume, the balance is somewhat more even, as can be seen from the token count.

The genuine messages are sourced from fifteen friends and colleagues and represent a wide range of topics, both personal and commercial in nature. The spam messages are sourced from sections 10-29 of the *spamarchive*[3] collection, as well as a batch of spam collected by the author and compatriots. The messages are from roughly the same time period (predominantly 2002-2003), with the genuine messages more widely time distributed, while the spam messages represent the more recent instances in circulation at the point the corpus was constructed.

Relevant information is extracted from the raw email data and marked up in XML. Retained fields include: *Date*, *From*, *To*, *Subject*, *Content-Type* and *Body*. Non-text attachments are discarded, though the meta-level structure is preserved. If an email consists of multiple sections, these are represented by <PART> tags with a *type* attribute specifying the section type.

Standard and embedded text is identified and marked up in XML with the tags <TEXT_NORMAL> and <TEXT_EMBEDDED> respectively. Embedded text is recognised via the '>' marker, with up to four nested levels of embedding.

Releasing personal, potentially confidential email data to the academic community requires an anonymisation procedure to protect the identities of senders and recipients, as well as those of persons, organisations, addresses etc. referenced within the email body. We use the RASP part-of-speech tagger (1.3) as well as finite-state techniques to identify and anonymise proper names, numbers, email addresses and URLs. The following tokens are used in place of their respective references:

- &NAME (proper names)
- &CHAR (individual characters)
- &NUM (numbers)
- &EMAIL (email addresses)
- &URL (internet urls)

The *From* and *To* fields contain the email addresses of the sender and recipient(s) respectively. We retain only top level domain (TLD) information from each field. For US-based sites, the TLD is defined as the string of characters trailing the final dot, i.e. 'com' in 'joe@yahoo.com'. For non-US sites, it is defined as the final 2-char country code, along with the preceding domain type specification, i.e. 'ac.uk' in 'joe@dur.ac.uk' or 'co.uk' in 'freecomputers@flnet.co.uk'. This allows for potentially useful analysis of high-level sending and receiving domains, without any individual identity traceability.

After applying the automatic anonymisation procedures, all of the genuine messages were manually examined by the author and a colleague to anonymise remaining sensitive references. This took a significant amount of time, but resulted in a consensus that the

[3] http://www.spamarchive.org

```
<MESSAGE>
<FROM> net </FROM>
<TO> ac.uk </TO>
<SUBJECT>
<TEXT_NORMAL> ^ Re : Hello everybody </TEXT_NORMAL>
</SUBJECT>
<DATE> Tue, 15 Apr 2003 18:40:56 +0100 </DATE>
<CONTENT-TYPE> text/plain; charset="iso-8859-1" </CONTENT-TYPE>
<MESSAGE_BODY>
<TEXT_NORMAL>
^ Dear &NAME ,
^ I am glad to hear you 're safely back in &NAME .
^ All the best
^ &NAME
^ - On &NUM December &NUM : &NUM &NAME ( &EMAIL ) wrote :
...
</TEXT_NORMAL>
</MESSAGE_BODY>
</MESSAGE>
```

Figure 4.1: *GenSpam* representation

data was sufficiently anonymous to be publicly released. A more detailed investigation into the anonymisation problem is carried out in Chapter (5).

It is to be expected that spam filtering with anonymised data is somewhat more challenging than it would be otherwise, as potentially useful information is necessarily lost. However, our experiments with both anonymised and unanonymised versions of *GenSpam* suggest that using unanonymised data results in only marginally better performance (around 0.003 improvement in recall), and that the difference between classification performance on anonymised and unanonymised data is not sufficient to cause concern about misrepresenting the task.

Figure 4.1 gives an example of the *GenSpam* email representation in XML format. The corpus is divided as follows:

- *Training set*: 8018 genuine, 31235 spam
- *Adaptation set*: 300 genuine, 300 spam
- *Test set*: 754 genuine, 797 spam

We source the *Adaptation* and *Test* sets from the contents of two users inboxes, collected over a number of months (Nov 2002–June 2003), retaining both spam and genuine messages. We take this approach rather than simply extracting a test set from the corpus as a whole, so that the test set represents a real-world spam filtering instance. The 600 messages making up the adaptation set are randomly extracted from the same source as the test set, facilitating experimentation into the behaviour of the classifier given a small set of highly relevant samples and a large background corpus.

4.4 Classification Model

4.4.1 Introduction

We use the following terminology and definitions:

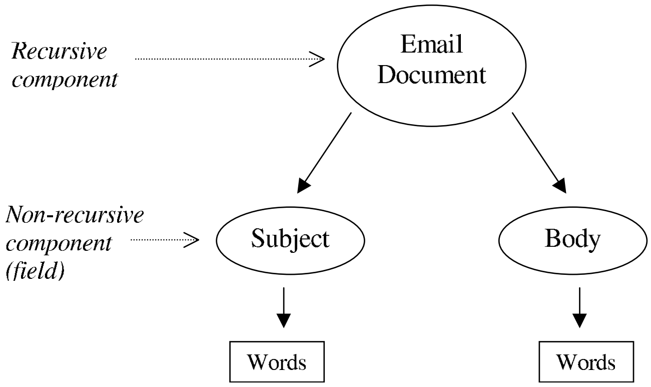


Figure 4.2: Example of semi-structured document

- *Document*: a discrete item of information (i.e. a single email message).
- *Token*: an atomic unit within a document.
- *Class*: a well-defined (possibly infinite) set of documents.

A semi-structured document is a singly-rooted tree (see Fig. 4.2). Non-leaf nodes represent structural document sections and leaf nodes represent content bearing sections.

The classification model we present is an *interpolated generative model*. That is, non-leaf (structural) node posterior probabilities are computed as an interpolation of sub-node posteriors, while leaf (content) node posteriors are estimated in the traditional generative fashion. The interpolation weights are optimised under the discriminative classification function; consequently the model bears some relation to the class of *hybrid generative/discriminative* classifiers, (Raina, Shen, Ng & McCallum 2004). By incorporating smoothed higher-order n-gram language models[4], local phrasal dependencies are captured without the undesirable independence violations associated with mixing higher and lower-order n-grams in a pure Naïve Bayesian framework (Tan et al. 2002). Additionally, through the use of interpolation, we incorporate an efficient, well-studied technique for combining probabilities to exploit document structure.

Although we only consider application of the proposed classification model to the 2-class classification problem, it readily scales to the more general N-class problem.

4.4.2 Formal Classification Model

We make the following assumptions:

1. *A document belongs to exactly one class.* This is clearly appropriate for spam filtering, though it is in principle quite simple to extend the model to allow documents to belong to multiple classes.
2. *Classification is carried out within a single domain, and within that domain, all documents have the same structure.*

[4] We use n-grams for efficiency and simplicity, though more advanced LM technology could be investigated.

Given a set of documents $\mathbf{D}$ and a set of classes $\mathbf{C}$, we seek to discover a set of classifications of the type $D_i \rightarrow C_j$ for $i = 1 \ldots |\mathbf{D}|$ where j ranges from $1 \ldots |\mathbf{C}|$ (given assumption 1).

We use the standard Bayes decision rule to choose the class with the highest posterior probability for a given document:

$$Decide(D_i \rightarrow C_j) \quad \text{where } j = \arg\max_k [P(C_k|D_i)] \tag{4.1}$$

The posterior probability of a non-leaf document node is calculated as a weighted linear interpolation of the posteriors of its N sub-nodes:

$$P(C_j|D_i) = \sum_{n=1}^{N} \lambda_n \left[P(C_j^n|D_i^n) \right] \tag{4.2}$$

where

C_j^n is the nth sub-component of class C_j

D_i^n is the nth sub-component of doc D_i

λ_n is the nth sub-component weight

An *interpolation scheme* is used to determine values for the λ's (see subsection 4.4.5).

Leaf-node posteriors are computed via *Bayes Rule*:

$$P(C_j^n|D_i^n) = \frac{P(C_j^n) \cdot P(D_i^n|C_j^n)}{P(D_i^n)} \tag{4.3}$$

C_j^n represents a specific leaf node within class C_j, and D_i^n the corresponding node within the document. Under the structure uniformity assumption (2), these are necessarily equivalent.

$P(C_j^n)$ is the *prior probability* for the node in question. We take all node priors within a given class to be equal to the class prior, i.e. $P(C_j)$.

The document node prior, $P(D_i^n)$, is constant with respect to class and thus often ignored in Bayesian classification models; however, valid interpolation requires true probabilities; thus we retain it. This carries the additional benefit of normalising for imbalanced field lengths. For instance, the amount of text in the *subject* field is usually significantly less than in the *body* field and therefore the class conditional likelihood for the *body* field will be disproportionately lower. However, scaling the class-conditional likelihood of each by the document node prior, which is multiplicatively proportional to the length of the field, counteracts the imbalance.

$P(D_i^n)$ can be expanded to

$$\sum_{k=1}^{|\mathbf{C}|} P(C_k^n) \cdot P(D_i^n|C_k^n)$$

which is the sum over all classes of the prior times the class-conditional likelihood for the given field.

$P(D_i^n|C_j^n)$ is the *language model probability* of the field D_i^n given C_j^n. In other words, it is the likelihood that the LM chosen to model field C_j^n generated the sequence of tokens comprising D_i^n.

For our experiments we use *n-gram* LM's. The n-gram model is based on the assumption that the existence of a token at a given position in a sequence is dependent only on the previous $n-1$ tokens. Thus the n-gram LM probability for a K-length token sequence can be defined (with allowances for the initial boundary cases) as

$$P_N(t_1, \ldots, t_K) = \prod_{i=1}^{K} P(t_i|t_{i-n+1}, \ldots, t_{i-1})$$

The formula is specialised for $n = 1, 2, 3 \ldots$

4.4.3 LM Construction

We adopt the basic formalisation for higher-order n-gram smoothing introduced by Katz (1987). This approach has been shown to perform well across a number of recognised data sets (Chen & Goodman 1996), and is widely used in mature language modelling fields such as speech recognition. In the bigram case, the formula is as follows:

$$P(t_j|t_i) = \begin{cases} d(f(t_i,t_j))\frac{f(t_i,t_j)}{f(t_i)} & \text{if } f(t_i,t_j) \geq C \\ \alpha(t_i)P(t_J) & otherwise \end{cases}$$

where

f is the frequency-count function

d is the discounting function

α is the back-off weight

C is the n-gram cutoff point

For higher-order n-grams the same principles are applied to form a *back-off chain* from higher to lower-order models. The n-gram cut-off point, C, is the threshold below which the observed number of occurrences is too low to draw reliable statistics from. The discounting function, $d(r)$ is used to remove some of the probability mass from those events that have been observed in the training data, thus making it available to unobserved events. The discounted probability mass is then distributed over lower-order distributions with the back-off weight insuring conformance to the probability model, i.e. $\alpha(w_i) = 1 - \sum \hat{P}(*|w_i)$ where $\hat{P}$ is the *discounted* bigram probability. A small probability must also be assigned to events that remain unobserved at the end of the back-off chain, i.e. unigram entries that have not been seen at all in the training data. We can use this to model the likelihood of encountering unknown words, given a particular class of documents.

Various discounting schemes have been proposed in the literature (Chen & Goodman 1996); we implemented *linear* and *Good-Turing* for our experiments, as well as two variants of a new discounting function, which we will call *confidence discounting*, based on the intuition that the amount of probability mass discounted from a given N-gram entry should be inversely proportional to the *confidence* we have in that particular entry (within certain boundaries), represented by the absolute number of times the entry was observed in the training data. This idea can be formulated as follows:

$$d(r) = \frac{r}{R}(\omega - \phi) + \phi \tag{4.4}$$

where

$$\begin{aligned} R &= \text{the number of distinct frequencies} \\ \phi &= \text{floor for lowest confidence} \\ \omega &= \text{ceiling for highest confidence} \end{aligned}$$

The value returned by the function ranges from ϕ to ω. R is an estimate of the highest level of confidence, chosen as the number of distinct N-gram frequencies because of its robustness to outliers. ϕ is chosen to represent the quantity of probability mass retained in the case of least confidence, and ω is chosen to represent the quantity of probability mass retained in the case of highest confidence (i.e. when the N-gram count approaches R). Note that when r exceeds R, an adjustment may need to be made to ensure the function does not return a value greater than one. The function is linear in the space r by $d(r)$.

A non-linear version can be formulated as follows:

$$d(r) = \frac{r(R-1)}{\frac{R}{\omega}(r-1) + \frac{1}{\phi}(R-r)} \tag{4.5}$$

In both cases, the values of the constants ϕ and ω can either be estimated autonomously from the data, or manually, based on empirical analysis. For our experiments we estimate ϕ from the training data, and use the LM-dependent value $1 - n_3/T$ for ω (where n_3 is the number of N-grams occurring 3 times, and T the total number of words encountered).

The assumption behind the non-linear form (4.5) is that confidence in a given N-gram should increase significantly after it occurs the first few times, and then continue to increase at a slower rate as it is encountered more and more often. The justification for this assumption is that as if an N-gram has been seen more than a few times (say around 5-10) it is likely to be more than just an erroneous or exceptional case, and our confidence in it should increase rapidly. However, once an N-gram has been seen many times already, seeing it a few more times should not imply such a significant increase in confidence.

4.4.4 Adaptivity

A realistic classification model for spam filtering should take account of the fact that spam evolves over time. It should also account for the fact that each individual spam filtering instance will have its own characteristics, due to the variation in email usage, but at the same time much evidence about the nature of spam versus genuine email will be common across all (or at least most) instances. In light of this we extend our model to incorporate both a *static* and *dynamic* element. The static element represents evidence contributed by

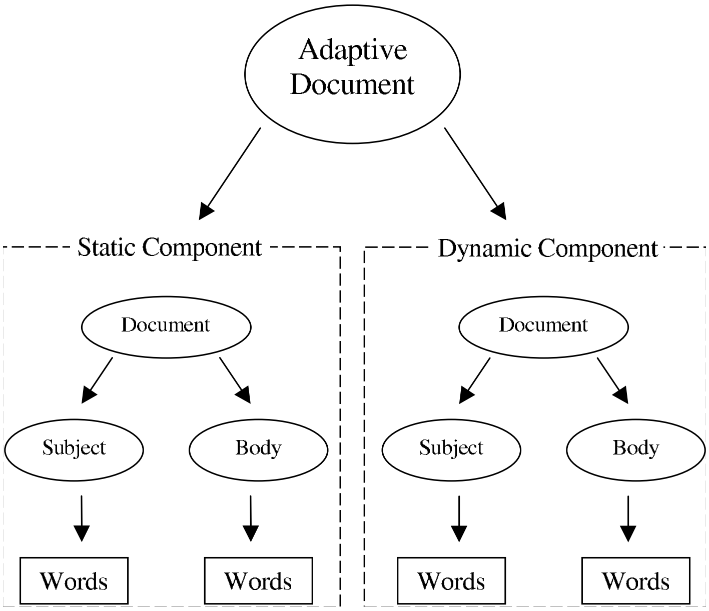


Figure 4.3: Example of adaptive document structure

LMs trained on a large background corpus, while the dynamic element represents smaller, instance-specific evidence from LMs that are regularly retrained as new data is accrued.

The decision rule (4.1) is expanded to:

$$\begin{aligned} &Decide(D_i \rightarrow C_j) \text{ where} \\ &\quad j = \arg\max_k [\lambda_s P_s(C_k|D_i) + \lambda_d P_d(C_k|D_i)] \end{aligned} \tag{4.6}$$

The subscripts s and d denote the static and dynamic elements, which are separate but identically structured estimates, derived from the static and dynamic LMs respectively. The modified decision rule can be interpreted as adding a binary-branching recursive top-level node to the document structure with both branches structurally identical but using different sets of LMs (Fig. 4.3). The adaptive decision rule can thus be rewritten as:

$$Decide(D_i \rightarrow C_j) \quad \text{where } j = \arg\max_k [P(C_k^a|D_i^a)] \tag{4.7}$$

with the superscript a denoting use of the adaptive structure.

4.4.5 Interpolation

The purpose of an interpolation scheme is to optimise the weights of two or more interpolated components with respect to their performance on a given data set, under a specified objective function (Jelinek & Mercer 1980). In our case, a component is represented by the posterior probability for a particular tree node. We choose the classification function itself (under a suitable evaluation metric) as the objective function, which has the advantage of precisely reflecting the nature of the problem. On the negative side, the classification function is *non-differentiable*, thus optimality of the interpolation weights cannot be estimated with derivative-based optimisation techniques which converge to optimality in a reasonably efficient manner. Rather, we must use an approximation algorithm to achieve near-optimality. In our experiments we only interpolate two components (see 4.6.3) so a simple hill-climbing algorithm suffices. However if a greater number of fields were utilised, a more complex algorithm would be required.

To maintain efficiency, we estimate interpolation weights in a bottom-up fashion, propagating upwards through the structural tree rather than iteratively re-estimating throughout the whole structure.

4.5 Experimental Analysis – *LingSpam*

To provide a measure of how our spam filtering approach compares to others presented in the literature, we carry out an experimental analysis on the widely-used *LingSpam* corpus (Androutsopoulos et al. 2000).

4.5.1 Data

The *LingSpam* corpus is divided into ten sections, which we used for ten-fold cross-validation, in line with previous work by Androutsopoulos et al. (Androutsopoulos et al. 2000) and others. Within each email message, only two fields are present: *Subject* and *Body*.

4.5.2 Experimental Method

The classification approach using the model we propose consists of two basic phases: Firstly the language models are constructed from the training data, and secondly the decision rule (equation 1) is used to classify the test data. The results we present were derived from ten-fold cross validation on the *LingSpam* data (each fold consisting of nine training sections and one test section). We experimented with different language model types and present results for unigram and smoothed bigram models using three different discounting schemes:

- GT = Good-Turing/Linear mix
- CL = Linear Confidence (4.4)
- CN = Non-linear Confidence (4.5)

The values of the interpolation (λ) weights were estimated using an interpolation scheme, as described above (4.4.5), and the values for the LM parameters C and ϕ (for the confidence discounting schemes) were estimated from the training data in a similar manner.

Our experimental method is as follows:

1. Choose LM type

2. For each cross validatory section, $S_i \in \{S_1, \ldots, S_{10}\}$

   Construct training set: $T_i = \{S_1, \ldots, S_{10}\} \setminus \{S_i\}$

Table 4.1: Comparative results on *LingSpam* corpus.

	GEN			SPAM		
Classifier	Recall	Precision	F1	Recall	Precision	F1
NB	-	-	-	82.35	**99.02**	89.92
kNN	-	-	-	88.60	97.40	92.79
Stacking	-	-	-	91.70	96.50	93.93
TreeBoost	-	-	-	97.92	98.33	98.12
LM (unigram)	99.09	99.51	99.29	97.55	95.51	96.52
LM (bigram GT)	99.65	**99.71**	99.68	**98.53**	98.27	98.40
LM (bigram CN)	99.77	99.67	99.72	98.35	98.84	98.59
LM (bigram CL)	**99.78**	99.67	**99.73**	98.35	98.91	**98.63**

Estimate parameters C_i and ϕ_i (confidence discounting only) using T_i

Estimate interpolation weights W_i using deleted interpolation scheme on T_i

Construct language models LM_i from T_i using C_i and ϕ_i

Classify S_i using LM_i and W_i, yielding results R_i

Calculate evaluation measures on R_i, yielding E_i

3. Calculate evaluation measure average over $\{E_1, \ldots, E_{10}\}$

4.5.3 Evaluation Measures

We report precision (p), recall (r) and F_1 for both classes - SPAM and GEN, defined in the usual manner:

$$precision(C) = \frac{\mathrm{TP}^c}{\mathrm{TP}^c + \mathrm{FP}^c} \qquad recall(c) = \frac{\mathrm{TP}^c}{\mathrm{T}^c} \qquad \mathrm{F}_1(C) = \frac{2 \times p(C) \times r(C)}{p(C) + r(C)}$$

where

TP^c is the number of true positives for class c.

FP^c is the number of false positives for class c.

T^c is the number of documents in c.

4.5.4 Results

For purposes of comparison, we present results on the *LingSpam* corpus from four other classifiers presented in the literature:

- **NB.** We include the best results reported by Androutsopoulos et al. (2000) for the Naïve Bayesian approach, using a lemmatized version of the *LingSpam* corpus and the *mutual information* (MI) metric for feature selection. They find NB to perform optimally in this case with a feature set of around 100 elements.

- **k-NN variant.** From the same paper, we include the best reported results for a variant of the k-nearest neighbour algorithm. As for NB, they perform feature selection based on the MI metric, and achieve optimal results with a smaller feature set of 50 elements.

- **Stacking.** This approach combines NB and k-NN in a *stack* of classifiers. Sakkis, Androutsopoulos, Paliouras, Karkaletsis, Spyropoulos & Stamatopoulos (2000) experiment with various configurations. We include the best reported results from their paper.

- **Boosting Trees.** We include the best results reported by Carreras & Marquez (2001) using several variants of the *AdaBoost* algorithm, based on learning and combining *weak rules* in a decision tree structure. They experiment with various tree depths using up to 2500 rules, and report enhanced performance comparative to previous work.

We report results for both classes, whereas previous studies have not reported performance on the GEN class. The reason for this is that because the classification task is binary, GEN *recall* is directly proportional to SPAM *precision* and GEN *precision* is directly proportional to SPAM *recall*. We report both classes for completeness. Table 1 displays results obtained using our classifier, alongside those previously published using the above classifiers. The bigram language model classifier with linear confidence discounting yields an approximate 30% reduction in error rate over the best of the previously published figures, however due to extremely narrow misclassification margins this difference is not statistically significant (according to a sign test).

4.6 Experimental Analysis – *GenSpam*

We now present an experimental analysis of the performance of a variety of classification techniques on the *GenSpam* corpus.

4.6.1 Classifier Comparison

In these experiments, we benchmark the *GenSpam* corpus by comparing our classification model with three popular alternatives: multinomial naïve Bayes (MNB), support vector machine (SVM) and Bayesian logistic regression (BLR). We refer the reader to 3.2.3 for a brief introduction to MNB and SVM.

BLR

Bayesian logistic regression is a relatively recent technique that falls into the family of regression methods for classification. A prior over feature weights is used to prefer sparse classification models and thus avoid overfitting and increase efficiency. Such a model was shown to perform competitively with the state-of-the-art on various TC datasets in (Genkin et al. 2005).

Training Data	Classifier	GEN recall	SPAM recall	accuracy
Training	MNB	0.9589	0.9322	0.9452
	SVM	0.9005	0.9837	0.9433
	BLR	0.8926	0.9862	0.9407
	ILM Unigram	0.9496	0.9674	0.9587
	ILM Bigram	0.9735	0.9636	**0.9684**
Adaptation	MNB	0.9682	0.9335	0.9504
	SVM	0.9854	0.9724	**0.9787**
	BLR	0.9642	0.9737	0.9691
	ILM Unigram	0.9775	0.9373	0.9568
	ILM Bigram	0.9682	0.9649	0.9665
Combined	MNB	0.9629	0.9297	0.9458
	SVM	0.9310	0.9887	0.9607
	BLR	0.9244	0.9887	0.9574
	ILM Unigram	0.9907	0.9674	0.9787
	ILM Bigram	0.9854	0.9737	**0.9794**

Table 4.2: *GenSpam Test* set results (best results for each dataset in bold)

4.6.2 Implementation

We will henceforth refer to our classification model as ILM (Interpolated Language Model), which we have implemented in perl. We have also implemented our own version of MNB following the standard model (McCallum & Nigam 1998), and use SVM^{light} (1.3), reporting results for the best performing linear kernel. We use the open source implementation of Bayesian logistic regression, BBR (Bayesian Binary Regression) provided by Genkin et al. (2005).[5]

4.6.3 Experimental Method

We use held-back sections of the training data to tune the ILM hyperparameters: unseen term estimates, n-gram cutoff and interpolation weights, as well as the regularization parameter in SVM^{light}. MNB doesn't have any hyperparameters, and BBR has an inbuilt '–autosearch' parameter to optimise the prior variance via 10-fold cross validation. We then evalute each of the classifiers on the test data in three sets of experiments, using as training data:

1. Just the *Training* data
2. Just the *Adaptation* data
3. A combination of both

4.6.4 Data

Our experiments make use of only two email fields - *Subject* and *Body*. These are of primary interest in terms of content, though other fields such as *From*, *To*, *Date* etc. are also of potential use. This is an avenue for further research.

[5] http://www.stat.rutgers.edu/ madigan/BBR/

We pre-process the corpus by removing punctuation and tokens that exceed 15 characters in length. We do not carry out stopword removal as it had a significantly detrimental effect on performance, especially in the SVM case. This is presumably due to the fact that stopword usage differs between spam and genuine email, and exemplifies the disparity between spam filtering and traditional text categorization tasks such as topic classification.

The ILM and MNB classifiers do not require scaling or normalisation of the data. For SVM and BLR, we construct L_2-normalised *tf∗idf* weighted input vectors, as used in previous studies (Joachims 1998).

4.6.5 Evaluation Measures

The binary classification task is often evaluated using the *accuracy* measure, which represents the proportion of instances correctly classified:

$$accuracy = \frac{\mathrm{TP}}{\mathrm{T}}$$

where TP is the number of true positives and T the total number of documents. We also report *recall* for each class separately, defined as before (4.5.3).

Assessing the recall performance of the classifier on spam and genuine email separately is important in the area of spam filtering, where high recall of genuine messages is of utmost importance. This imbalance in the nature of the task necessitates evaluation schemes that recognise the asymmetric cost of misclassification (4.6.7).

4.6.6 Hyperparameter Tuning

We varied certain features of the ILM classifier and observed results on held-back sections of the training data to determine the better-performing configurations. The results led us to draw a number of conclusions:

- We use only unigram and bigram language models, as higher order n-gram models degrade performance due to excessive sparsity and over-fitting.

- Intuitively, we might expect spam to contain more unknown words than genuine email, due to the additional lexical noise. The LM unseen event probability can be used to model this phenomenon. We optimise unseen event probabilities empirically from held out sections of the training data, and arrive at the following values:

Unigram	GEN	1×10^{-8}
	SPAM	1.2×10^{-8}
Bigram	GEN	1×10^{-8}
	SPAM	1×10^{-7}

- The discrepancy in LM size between different classes as a result of unbalanced training data can lead to classification errors because parameters in larger LMs receive proportionally less of the overall probability mass. This is especially noticeable in higher-order LMs where the potential feature space is much larger. One method for countering this is to raise the n-gram cutoff point (see 4.4.3) for the larger class.

> We call this technique *LM balancing*, and found it to have a positive effect on performance for bigram LMs. Hence, we use C=1 for GEN and C=2 for SPAM in the body field LMs generated from the *Training* dataset, and C=1 for all other LMs.

After tuning on held-back sections of the training data, we use the linear kernel and choose the value C=1 for the regularization parameter in SVM^{light}. We use a Gaussian prior distribution and the '–autosearch' parameter in BBR to optimise the prior variance via 10-fold cross validation.

4.6.7 Results and Analysis

We present results for the various classifiers on the *GenSpam* corpus under symmetric and asymmetric evaluation schemes.

Symmetric Classification

Table 4.2 displays the performance of the classifiers on the *Test* dataset under the standard symmetric evaluation scheme. For the *Combined* results we merge the *Training* and *Adaptation* sets in the case of MNB, SVM and BLR, and combine them by the adaptive decision rule (4.6) for ILM.

The amount of adaptation data is too small to reliably estimate interpolation weights for the adaptive decision rule. In practice, therefore, we would set these manually. Given that the distribution of interpolation weights can be interpreted as a probability distribution with each weight representing the probability that a particular component contains relevant information, we choose the distribution that is most uncertain, governed by the principle of *maximum entropy.* Without any prior knowledge about the optimal weight distribution, this equates to balancing the component weights.

As expected, MNB is highly efficient, but performs somewhat worse than the best performing model in each category.

The SVM classifier performs well when trained only on the *Adaptation* data, but relatively poorly when trained on the *Training* data. This is because the *Adaptation* set has certain properties that suit the SVM model: the distribution of the training data matches that of the test data (they are both roughly balanced), the data is linearly separable and the diminutive number of training samples allows the wide-margin effect to have a significant impact. Conversely, the *Training* set does not particularly suit the SVM model: the training distribution does not match the test distribution and the training data is unbalanced and non linearly separable. It has been shown empirically that the SVM model chooses a suboptimal decision boundary in the presence of divergence between the training and test distributions (Forman & Cohen 2004) and this is supported by our results.

The results for BLR are quite similar to SVM in these experiments, though slightly inferior. Estimation of the prior variance by cross validation does improve performance, though it dramatically increases training time.

The ILM classifier performs competitively across the board, and particularly when the adaptive decision rule is used. Figure 4.4 plots classification performance as a function of the adaptive interpolation component weight, so that $x = 0$ represents only the *Training* models and $x = 1.0$ represents only the *Adaptation* models. For both unigram and bigram

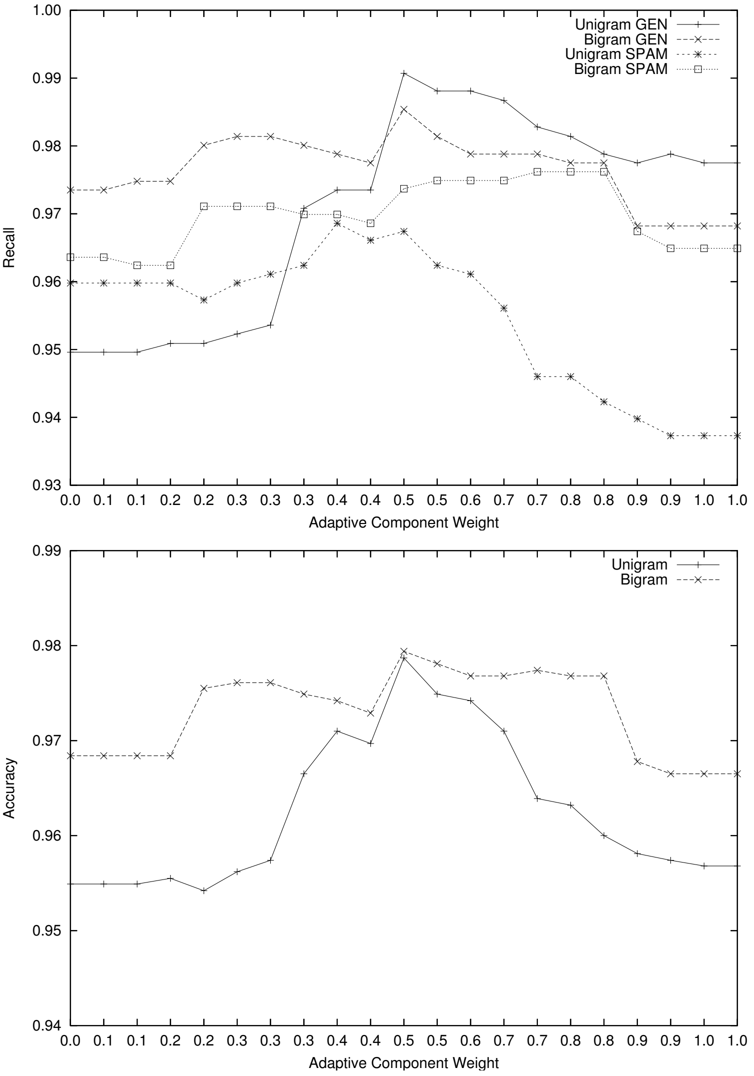


Figure 4.4: ILM *recall* (GEN and SPAM) and *accuracy* under adaptive weight interpolation

LMs, the ILM classifier benefits from the combined estimates; however the benefit is most significant in the unigram case. It is interesting to note that both classifiers reach a performance peak at the point where the static and dynamic weights are balanced, i.e. when there is an equal contribution from both models.

Asymmetric Classification

While the symmetric results are informative, they do not present a realistic view of the spam filtering problem, in which the correct classification of genuine mail is of much greater import than the occasional misclassification of spam. There are a number of ways to evaluate spam filters in the presence of asymmetric misclassification cost; we will use a scenario in which a predefined recall threshold for genuine mail must be reached by the classifier. We set this threshold at recall=0.995 i.e. we allow, on average, no more than one genuine message in every 200 to be misclassified.

We control the bias in the MNB, SVM and BLR classifers by adjusting the decision

Training Data	Classifier	GEN recall	SPAM recall	accuracy
Training	MNB	0.9960	0.1556	0.5642
	SVM	0.9960	0.7064	0.8472
	BLR	0.9960	0.8105	0.9007
	ILM Unigram	0.9960	0.7340	0.8614
	ILM Bigram	0.9960	0.8331	**0.9123**
Adaptation	MNB	0.9960	0.4090	0.6944
	SVM	0.9960	0.9147	0.9491
	BLR	0.9960	0.9097	**0.9542**
	ILM Unigram	0.9960	0.8269	0.9091
	ILM Bigram	0.9960	0.8934	0.9433
Combined	MNB	0.9960	0.4103	0.6950
	SVM	0.9960	0.8808	0.9368
	BLR	0.9960	0.9021	0.9478
	ILM Unigram	0.9960	0.9573	0.9761
	ILM Bigram	0.9960	0.9674	**0.9813**

Table 4.3: Asymmetric results (best results for each dataset in bold)

threshold at a granularity of 0.001. The SVM model can also be biased by increasing the misclassification cost for a given class (-j option in SVM^{light}); however we found that even for highly skewed values of this parameter (ratio of 1/1000) the requisite genuine mail recall threshold remained unreached.

The language modelling aspect of the ILM classifier allows various ways of biasing of the model in favour of a given class. We control the bias by reducing the unseen term estimate for the SPAM body LMs until the genuine threshold is reached.

Table 4.3 displays the results of biasing the models to reach the genuine mail recall threshold. The full ILM model, trained on the combined static and dynamic data, significantly outperforms any of the other classifiers, with the accuracy of the bigram variant actually increasing as the genuine recall threshold is reached. This suggests that the ILM classifier is well suited to the spam filtering task.

Figure 4.5 plots the receiver operator characteristic (ROC) curves for each of the best-performing classifier configurations (BLR and SVM – adaptation data; ILM – combined data). This provides further evidence of the effectiveness of biasing the language models by adjusting unseen word estimates, as opposed to biasing the decision boundary. Both the unigram and bigram ILM classifiers, when trained on the combined data, are able to maintain high genuine recall values without sacrificing overall accuracy.

4.7 Discussion

Interpolating LM-based structural components provides a natural way to efficiently combine estimates from different distributions. With n-gram LMs, the classifer uses efficient maximum likelihood estimation and hence has a training and classification time complexity roughly linear in the input size. However, an approach such as the one presented in this study has its drawbacks, as it requires estimates for a significant number of hyperparameters. These must be derived either empirically or by potentially expensive cross

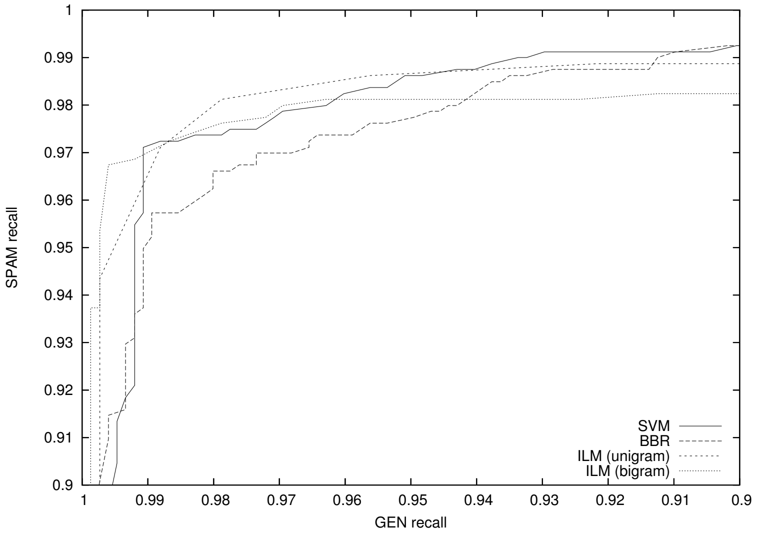


Figure 4.5: Classification ROC curves

validation. The parametricity of the ILM model also makes it potentially sensitive to changes in the nature of the problem domain, a relevant issue when dealing with the ever-changing nature of spam, and email in general. An obvious line of future research is to investigate methods for estimating the ILM hyperparameters both robustly and efficiently.

Bearing in mind the success of the ILM classifier at combining evidence from distinct training distributions, it would be interesting to investigate analagous techniques for discriminative models such as the SVM and BLR. A possible starting point would be to examine the effects of combining judgements from separate SVM or BLR models trained on distinct data. An interpolative method could potentially be used in this setting, which would not harm the tractability of the base classifier, though it would introduce new hyperparameters. A further avenue of research is to investigate alternative methods of biasing the discriminative classifiers to improve their asymmetric performance. One possible approach would be to investigate recent research into utilising uneven margins in the SVM model (Li, Bontcheva & Cunningham 2005). This technique has shown some promise when dealing with skewed training data, though it has not been examined in the context of handling asymmetric classification costs.

4.8 Conclusions

We have presented a new corpus of genuine and unsolicited email, *GenSpam*, which we hope will aid in providing opportunity for more realistic spam filtering experiments and ultimately enhance efforts to build more effective real-world spam filters. Obtaining spam is relatively easy, and a potentially important task for the future is to update the corpus with more recent spam, improving its relevance. We believe that the anonymised genuine email content represents a significant contribution in itself, and may be useful for a wider range of NLP tasks than just spam filtering.

We have also presented an efficient, adaptive classification model for semi-structured documents that extends similar work in the semi-structured and hybrid generative/discriminative classification fields. We demonstrate that our classifier is effective at combining evidence from distinct training distributions (an important attribute for adaptive classification), and experiments on *GenSpam* suggest that the model is well suited to spam filtering, maintaining high levels of genuine recall without loss of overall accuracy.

The work presented in this chapter, in particular with regard to the construction of the *GenSpam* corpus, was carried out a short while prior to the establishment of the TREC (text retrieval conference) spam track.[6] A significant and concerted effort was made by the organisers of this event to develop a realistic content-based spam filtering test suite to facilitate model comparison, and this culminated in the release of the TREC spam filtering corpus (Cormack & Lynam 2005). During construction of this corpus, we were contacted by the TREC spam track organisers, who were interested in investigating whether *GenSpam* could be included as part of the test suite for the TREC spam track. Unfortunately it turned out that certain header information, discarded by necessity during the construction of *GenSpam* due to donor agreements, rendered the data incompatible with the TREC spam track corpus and so it could not be incorporated.

In recent years, spam filtering has become a multi-billion dollar business[7] and commercial organisations specialising in fighting spam have access to ever-growing databases containing billions of messages, both spam and genuine. These resources cannot be matched in the academic community and it is our opinion that mainstream spam filtering research has moved, and will continue to move further, out of the academic realm. Arguably, there is still scope for academic research in specific 'niche' areas, such as, for instance, investigating techniques for dealing with specific types of spam; however, the nature of spam changes so rapidly that academic research often becomes obsolete before the peer-review process can allow it to be published. The TREC spam track closed in 2007, though academic spam filtering research continues to be published in arenas such as the Conference on Email and Anti-Spam (CEAS)[8] which attempts to pull together research from both commercial and academic communities. Whatever the future of spam filtering, it is unarguably the case that academic spam filtering research along the lines presented in this study has contributed to the effectiveness of current commercial spam filtering technology.

[6] *http://plg.uwaterloo.ca/ gvcormac/spam/*

[7] *http://www.itsecurity.com/features/real-cost-of-spam-121007*

[8] *http://www.ceas.cc/*

Chapter 5

Anonymisation

In this chapter we consider binary classification for automatic *anonymisation* of written text, an under-researched NLP task of particular importance in the context of data sharing. We introduce the anonymisation task, paying attention to specificities such as evaluation and validation. We discuss the method by which we have constructed a new, publicly-available corpus of email data for evaluating anonymisation systems and report initial results for the task using an off-the-shelf state-of-the-art HMM-based tagger and an alternative interactive binary classifier based on a probabilistic model defined over syntax trees. This chapter is expanded from work introduced in the proceedings of the fifth international conference on language resources and evaluation – *LREC 2006* (Medlock 2006*b*).

5.1 Introduction

Statistical NLP requires training data from which to derive model parameters, and test data on which to execute and evaluate techniques. If such data is shared between researchers, comparisons of different approaches may be reliably drawn. However, this can be problematic if the data involved is sensitive in nature (eg. personal email). In such cases, measures must be taken to obscure the identities of real-world entities revealed in some way by the text. In some cases, entities will be referenced directly, while in others, indirect but related references may betray their identity. The nature of these references will vary depending on the characteristics of the text, and whether a given reference is sensitive clearly requires a measure of subjective judgement. In some cases, the identity of the author of a piece of text may be revealed through his/her use of language or writing style.

In this study we address the problem of obfuscating textual references to real world entities, which we will refer to as *anonymisation* (sometimes also called *obfuscation* or *deidentification*). This task is relevant not only in the case of data for NLP research, but also more widely in any area where textual data sharing is of benefit. For example, in the medical domain, information about the diagnosis and treatment of past patients can be used to inform current procedures and to establish statistical trends; however, such data often contains references to actual patients and must therefore be anonymised before it can be shared.

The cost of anonymising large data sets by hand is often prohibitively high. Consequently, data that could be widely beneficial for research purposes may be withheld to protect its authors against undesirable legal and personal repercussions. A potentially viable alternative to manual anonymisation is automatic, or semi-automatic anonymisation through the use of NLP technology, if the effectiveness of such a procedure can be reliably established.

The contributions of this study are as follows:

- We present a description of the anonymisation problem and consider how the characteristics of the task affect the manner in which it is approached.
- We present a new corpus of personal email text as a benchmark for evaluating and comparing anonymisation techniques, with particular attention given to the semi-automated *pseudonymisation* procedure used to prepare the corpus for public release and the two annotation schemes used to represent different levels of sensitivity.
- We discuss evaluation strategies.
- We report initial results using two classifiers: an off-the-shelf state-of-the-art HMM-based tagger and an alternative method based on a simple probabilistic model defined over syntax trees.

5.2 Related Work

There is little in the way of published literature on the topic of anonymisation in general, and no detailed studies of anonymisation methods using NLP technology. A number of articles have been written on privacy issues as they relate to the ethical storage and use of data (Clarke 1997, Corti, Day & Backhouse 2000). Additionally, some researchers have highlighted the need for anonymisation in the area of automatic data mining and knowledge discovery (Wahlstrom & Roddick 2001). Roddick & Fule (2003) propose a method for automatically assessing the sensitivity of mining rules which bears some relation to the task considered in this paper, though is not of direct relevance. Anonymisation has also been discussed in the context of the electronic storage of medical records (Lovis & Baud 1999) and in relation to various other public data repositories, eg. (ESDS 2004). Perhaps the most comprehensive study of anonymisation is carried out by Rock (2001). She considers many aspects of the problem, highlighting both the reasons why corpora anonymisation is important and the particular nuances of the task. The following issues (amongst others) are addressed:

- Prevalent attitudes towards anonymisation amongst linguistic researchers
- Potential personal and legal implications of publicly available unanonymised corpora
- Which references should be anonymised
- Options for replacing sensitive references

5.3 The Anonymisation Task

We define the anonymisation task in terms of the following:

- *token*: a whitespace-separated unit of text
- *document*: an ordered collection of tokens

removal:	*Jo Bloggs works at AVC Books*	⟶	<REF> *works at* <REF>
categorisation:	*Jo Bloggs works at AVC Books*	⟶	<PER> *works at* <ORG>
pseudonymisation:	*Jo Bloggs works at AVC Books*	⟶	*Si Day works at NWK Books*

Figure 5.1: Example of anonymisation processes

- *reference*: a span of one or more tokens used by the author to refer to a concept outside of the language
- *sensitivity*: a binary measure determining whether or not a particular reference, if publicly disclosed, might potentially cause harm or offence and thus engender undesirable personal or legal repercussions

Given these premises, we present the following definition:

> ***Anonymisation* is the task of identifying and neutralising sensitive references within a given document or set of documents.**

The task of anonymisation can be seen as a two-stage process. Firstly, sensitive references must be identified, and secondly they must be neutralised. In this context, neutralisation means obscuring the link provided by a given reference to a real-world entity by means of:

- *removal*: replacing a reference with a 'blank' placeholder
- *categorisation*: replacing a reference with a label in some way representing its type or category.
- *pseudonymisation*: replacing a reference with a variant of the same type

Figure 5.1 gives an example of each of these techniques. Note that the identification phase involves an implicit sensitive/non-sensitive classification as well as detection of the reference boundaries.

Because sensitive references are usually those that refer directly to real-world entities, our formulation of anonymisation is quite similar in nature to the task of Named Entity Recognition (NER) which has received significant attention in recent years, and we would expect similar ideas to find application in both areas. It might be appealing to consider anonymisation as a special variant of NER; however, the tasks are not strictly subsumptive:

- Sensitive references are not necessarily named entities. For instance consider the following sentence:

  John Brown, the long jump record holder, retired yesterday.

  The constituent phrase *long jump record holder* betrays the identity of the named entity *John Brown* and is therefore a sensitive reference, though it is not itself a named entity.
- NER operates on the basis of objective judgements about the nature of referent entities (*Cambridge* is a place) whereas anonymisation relies on subjective judgements about referential sensitivity (*Cambridge* may or may not be a sensitive reference).

- NER is the process of identifying and classifying entity references, whereas anonymisation can include removal or pseudonymisation.

The inherent subjectivity of anonymisation means that different instances of the task may exhibit different characteristics even within the same domain. In light of this, it is probably impractical to deploy a solution requiring a large amount of annotated training data, bearing in mind that such training data may not generalise within the same domain, let alone across domains. In reality, application of an NLP-based anonymisation procedure would probably be carried out on an instance-by-instance basis, with rapid adaptation to the characteristics of the required solution through the use of interactive, weakly-supervised machine learning techniques.

Another important factor when considering the application of previous research into NER to the anonymisation problem is that NER has traditionally been carried out in the newswire domain where quite strict grammatical and orthographic conventions are observed and where the range of entity references tends to be quite limited. Conversely, the data that we present as a testbed for anonymisation is informal email text, where the use of grammar and orthography is highly colloquial in nature and there is a wider range of entity references (see 5.4.3).

5.4 Corpus

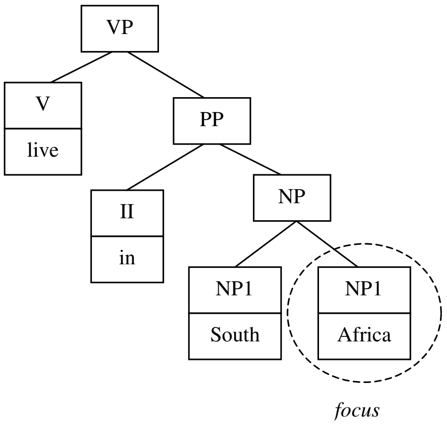
We have assembled a publicly-available[1] data set, dubbed ITAC (Informal Text Anonymisation Corpus), as a testbed for the anonymisation task.

5.4.1 Corpus Construction

The corpus is comprised of approximately 2500 personal email messages collected by the author over a seven-year period divided as follows:

- Training set: 666,138 tokens, pseudonymised, unannotated
- Test set: 31,926 tokens, pseudonymised, annotated
- Development set: 6,026 tokens, pseudonymised, annotated

The authorship of the text is highly varied, with both private and corporate communication represented, and the language and orthography consequently exhibits much variability. Capitalization and punctuation are often used inconsistently and in many cases are entirely absent, making reliable sentence boundary detection difficult. Though some automatic sentence boundary detection techniques were investigated and a significant amount of time was spent manually delimiting sentences, the final data set (especially the training data) still contains many spurious sentence boundaries. Additionally, the text contains many spelling and grammatical errors and inconsistencies. Whilst such issues increase the difficulty of the task, they are to be expected when working with informal text.



orthography:	First letter capitalised
part of speech:	“NP1”
parent constituent label:	“NP”
inner left constituent label:	“NP1”
2nd inner left constituent label:	∅
inner right constituent label:	∅
outer left constituent label:	“II”
outer left constituent token:	“in”

Figure 5.2: Feature set example

5.4.2 Pseudonymisation

Releasing data for the anonymisation task introduces an interesting conundrum: a realistic anonymisation testbed relies on sensitive experimental text with references preserved to facilitate the task, yet such text, in its original form, requires anonymisation before it can be publicly released. We overcome this problem by using a hybrid semi-supervised and manual *pseudonymisation* procedure to anonymise sensitive references without changing their nature. The procedure uses syntactic and orthographic features to cluster more obviously sensitive terms (such as person names) into semantically coherent groups and then randomly chooses replacement pseudonyms appropriate to the semantic category of the cluster, which is labeled manually. The text is then scanned manually to identify and pseudonymise more complex sensitive references.

We use the RASP parser (1.3) to generate the feature set for term clustering. The following syntactic and orthographic features are used:

- *part-of-speech*: a token’s part-of-speech, as assigned by the RASP PoS tagger
- *inner left constituent label*: the label of the focus constituent’s left sister
- *2nd inner left constituent label*: the label of the focus constituent’s left-but-one sister
- *inner right constituent label*: the label of the focus constituent’s right sister
- *outer left constituent label*: the label of the terminal constituent directly preceding the scope of the focus constituent’s immediate ancestor
- *outer left constituent token*: the surface form of the terminal constituent directly preceding the scope of the focus constituent’s immediate ancestor
- *orthography*: set of nine non-mutually exclusive orthographic features:
  - First letter capitalised (eg. *Mary*)
  - All letters capitalised (eg. *BARGAIN*)
  - Single capital letter (eg. *I*)

[1] http://www.cl.cam.ac.uk/users/bwm23

Cambridge	NP	city of Cambridge hencewith on	Bath/York/Cambridge/London/Leeds

Figure 5.3: Annotation example

- Integer-like number (eg. *01985*, token length also part of the feature)
- Float-like number (eg. *12.75*)
- Contains non-alphanum char (eg. *Yahoo!*)
- Contains period (eg. *S.W.A.T.*)
- Contains hyphen (eg. *26-year-old*)
- Contains an upper/lower case or alphanumeric mix (eg. *BigSplash*, *win2000*)

Figure 5.2 illustrates the use of these features via an arbitrary syntax tree fragment. Potentially sensitive terms with identical features are clustered, and each resulting cluster is presented to a human annotator, who classifies the whole cluster as either sensitive or non-sensitive and labels it with a semantic category if appropriate. An example of such a cluster is given in Figure 5.3.

Because many of the more obvious sensitive references appear in similar contexts, labeling an entire cluster saves much time over annotating individual examples. When a ceiling on annotation cost has been reached, the data is scanned using the information acquired through the annotation process and pseudonyms are automatically generated (by random selection from previously compiled gazateers) for all references that have been identified as sensitive and labeled with a semantic category. Pseudonyms are chosen under the constraint that a given term is always replaced by the same pseudonym. This preserves the distribution of sensitive terms across the corpus, an important characteristic of the data. The automatically generated pseudonyms are then propagated through the text to minimise the number of cases missed due to sparse feature sets.

Because of the nature of the text, only firstname, surname and certain location names can be safely pseudonymised by automatically generated replacements. Names of organisations, for instance, often contain terms that cannot be automatically pseudonymised without changing the concept conveyed. For example, *The Financial Times* must be replaced with a phrase that carries a similar conceptual idea, while obscuring the identity of the actual organisation. This is a subtly difficult task and cannot reliably be carried out automatically. Consequently we spent a number of days manually generating pseudonyms for such instances and scanning the entire corpus for other references that might betray the identity of actual entities.

An overview of the process is as follows:

- Parse text with RASP.
- Generate feature sets for potentially sensitive terms.
- Cluster terms by feature equivalence.
- Present clusters to human for sensitivity and type classification.
- Generate pseudonyms of same entity type for specified references.
- Propagate pseudonyms throughout text.
- Examine text for missed sensitive references and manually generate replacement pseudonyms.

Note that due to the predictable coding of email addresses, URLs and date/time references, we do not consider them as part of the anonymisation process for the purposes of this study; rather we identify and anonymise them beforehand using regular expression matching.

5.4.3 Annotation

In light of the subjectivity of the sensitivity measure, we use two annnotation schemes to represent different views of what constitutes a sensitive reference. In the first, which we will call *blanket* anonymisation, we label as sensitive every reference that could potentially be used to trace the identity of a person or organisation, even if the chance of undesirable personal or legal repercussions is small. References of the following nature are included:

- Person, organization and location names and descriptors
- Postal addresses and telephone/fax numbers
- Commercial titles (*Yahoo!*)
- Film, TV and book titles (*Star Wars*)
- Job titles (*Director of Graduate Studies*)
- Geographic/ethnic terms (*S. African*, *Hebrew*)
- Titles of academic papers (*A study of text classification methods*)
- Course titles (*Computer Science*)
- Conference titles (*5th Conference on Gene Identification*)
- Usernames/passwords
- Transactional identification/reference codes

This is not a definitive list, but covers most of the types of reference found in our corpus.

The second annotation scheme, which we will call *selective* anonymisation, involves labelling only those references which relate directly to a person or organisation and thus consitutes a minimum level of anonymisation. These include:

- Person and organization names and descriptors
- Postal addresses and telephone/fax numbers
- Commercial product names
- Usernames/passwords
- Transactional identification/reference codes

Whilst the risk of traceability may be increased under this scheme, reduced intrusion means advantageously less distortion of the data.

The *Development* and *Test* data sets are manually annotated using both schemes, while the training data is unlabeled. The current annotation schemes contain no entity class information, thus limiting experiments to the identification/removal variant of the anonymisation task. Class information could be added to the existing sensitivity annotation schemes, either by ourselves or others, and this would facilitate experimentation into the identification/classification variant of the task.

5.4.4 Format

The corpus is formatted on a one-sentence-per-line basis (though due to boundary detection errors, sentences are sometimes split over multiple lines). The data is tokenised using

```
From : " <ANON> Lance Malone </ANON> " ( &EMAIL )
To : " <ANON> tabitha ropp </ANON> " ( &EMAIL )
Subject : An email
Date : &DATE &TIME +0100
<ANON> Tabitha </ANON> ,
I can see absolutely no reason for your blank emails .
Can you see this one ?
I suppose you can because you 're reading this .
I 'VE FINISHED WORK ! ! ! ! !
I had a pretty hectic day today .
There was really too much to finish .
Still .
Have a relaxing weekend .
Doing anything interesting ?
<ANON> O </ANON>
```

Figure 5.4: Sample ITAC representation

the RASP tokeniser, which is based on a small number of regular expressions compiled using *flex*.[2] Orthography and punctuation are preserved as far as possible and codified references (such as email addresses) are represented by &*REF_TYPE* (eg. &EMAIL). Annotation is added in the form of <ANON> ... </ANON> tags that delimit sensitive references. Figure 5.4 shows a small sample from the blanket annotated version of the test data set.

5.5 Classification

5.5.1 Lingpipe

To provide baseline classification results on the new corpus, we use a hybrid first/second-order sequential HMM-based classifier called Lingpipe.[3] HMM-based techniques have proven successful for NER (Zhou & Su 2001) and *Lingpipe* has achieved state-of-the-art results on a number of well-known test corpora for the NER task in both the newswire and biomedical domain. We use the features automatically extracted by *Lingpipe* for the NER task, which are a combination of lexical, orthographic and contextual cues.

5.5.2 An Interactive Approach

Motivation

Different instances of the anonymisation problem may exhibit different characteristics even within the same domain, eg. a reference considered sensitive in one instance may not be considered so in another. In light of the subjectivity of the task, we propose that it makes sense to explore the use of interactive training techiques such as *active*

[2] http://dinosaur.compilertools.net

[3] www.alias-i.com/lingpipe

learning (Cohn, Ghahramani & Jordan 1995) to fit particular instances of the task with the minimum cost in terms of manual input.

We present a hybrid weakly-supervised / active learning approach using a rich feature space generated by a syntactic parser. The syntactic landscape is used both to identify reference boundaries and to discover generalising orthographic and syntactic patterns which can be used to rapidly fit the model to the particular characteristics of the task in question. Our learning strategy is based on active learning for tasks such as named entity recognition (NER) eg. (Shen et al. 2004) and also on studies that combine labeled and unlabeled data, eg. (McCallum & Nigam 1998). We extend this framework by allowing the annotator to make labeling decisions not about individual samples but about collections of samples in a particular syntactic context, hence increasing the efficiency of the learning process.

Formal Model

We define the identification phase of the anonymisation task as follows:

Given a body of text represented by a collection of sentences, $\{S_1, \ldots, S_n\}$ where each sentence consists of an ordered collection of tokens, $S_j = \{t_{j1}, \ldots, t_{jk}\}$, we seek to discover a set of sensitive references, R_j, for each sentence S_j, such that each $r \in R_j$ is a contiguous ordered subset of S_j.

After parsing, a sentence is represented as a syntax tree, with nodes representing syntactic *constituents*. Leaf nodes represent single token constituents, which we will call *terminal* constituents, while non-leaf nodes represent multi-token spans, which we will call *nonterminal* constituents. By representing a contiguous token span, each constituent is a potential reference, and because syntactic constituency reflects semantic reference, we expect constituent boundaries to reflect reference boundaries. Under this assumption, the task is to classify constituents as either sensitive or non-sensitive. To view this in a probabilistic framework, we estimate the probability that a given constituent belongs either to the class of sensitive or non-sensitive constituents, expressed as the posterior probability of class membership, $P(y|c)$, where $y \in${S,N}, and c is the given constituent (S, N denotes *sensitive*, *non-sensitive* respectively).

Each constituent c_i is assigned to the class with the highest posterior:

$$\arg\max_y [P(y|c_i)]) \tag{5.1}$$

The class posterior is calculated as a linear interpolation of two component distributions, one representing the constituent by a set of general orthographic and syntactic features, the other by its lexical form (in the terminal case) or its subconstituents (in the nonterminal case). We define two feature functions f_1 and f_2 over constituents to extract the features for each of the two distributions respectively. For terminal constituents, the class posterior is defined as::

$$\begin{aligned} P(y|c_i) =& \lambda P(y|f_1(c_i)) + (1{-}\lambda)P(y|f_2(c_i)) \\ & \text{where } 0 \leq \lambda \leq 1 \end{aligned} \tag{5.2}$$

The weights can be optimised to reflect the relative informativeness of the distributions, however we use balanced weights ($\lambda = 0.5$). f_1 and f_2 represent two 'views' on the data, which we will use when clustering the data for active learning. Note that the idea of separating the feature space into distinct views is explored by Blum & Mitchell (1998), Abney (2002) and others in the context of weakly-supervised learning via *co-training*. As discussed later, we use EM rather than co-training for parameter estimation at each training iteration, though co-training would be a potentially effective alternative. This is an avenue for future research.

Nonterminal constituents must be dealt with somewhat differently due to their hierarchical structure. The posterior estimate for a nonterminal constituent is given by:

$$P(y|c_i) = \lambda P(y|f_1(c_i)) + (1{-}\lambda)\frac{1}{M}\sum_{j=1}^{M} P(y|sc_{ij}) \tag{5.3}$$

where constituent c_i has M subconstituents, $\{sc_{i1}, \ldots, sc_{iM}\}$. This models a nonterminal constituent's sensitivity as an unweighted average of the sensitivity of its subconstituents. This is an oversimplification but it is efficient to compute and motivated by the plausible assumption that the sensitivity of a constituent will be governed largely by the individual sensitivity of its subconstituents.

Parameter Estimation

To fit the model, we use a combination of EM and active learning, in a manner similar to McCallum & Nigam (1998), utilising the rich syntactic feature space to cluster unlabeled instances and thus improve annotation efficiency. At each iteration, a batch of unlabeled sample clusters are chosen for annotation and added to the labeled pool. The labeled and unlabeled samples are then combined using EM to re-estimate the model parameters and the process iterates.

Unlabeled samples are chosen based on their *uncertainty* (Cohn et al. 1995) and *representativeness* (Tang, Luo & Roukos 2001, Shen et al. 2004). In probabilistic binary classification models such as ours, uncertainty can be defined as the absolute discrepancy between the class posterior estimates, given a particular sample:

$$\mathrm{unc}(c_i) = |P(S|c_i) - P(N|c_i)|$$

where $\mathrm{unc}(c_i)$ ranges from 0 (most uncertain) to 1 (most certain).

Sample representativeness is captured by clustering and ranking samples according to cluster size. We introduce the concept of a *margin of uncertainty* (MoU) within which the largest (most general) clusters are sought. This can be seen as an efficient approximation to finding a maximum of the combined uncertainty and representativeness of a given sample, and is defined as follows:

$$\mathrm{MoU}(c_i) = \begin{cases} \text{True} & \mathrm{unc}(c_i) \leq \omega \\ \text{False} & \mathrm{unc}(c_i) > \omega \end{cases} \tag{5.4}$$

where ω is a parameter governing the width of the margin, ranging between 0 and 1.

The largest clusters are examined to find their most representative samples, which are presented to the annotator along with a condensed list of the other constituent phrases in the cluster (see Fig. 5.3). The annotator is then asked to judge whether the selected

Given:
current training distribution parameters Θ
parsed document collection $\mathbf{D}$
number of annotations per cycle n
Loop until input cost ceiling reached:

- Extract consituent collection $\mathbf{C}^{all}$ from $\mathbf{D}$ such that:

$$\forall c_1, c_2 \in \mathbf{C}^{all}[\neg\text{subsumes}(c_1, c_2)] \tag{5.5}$$

- Choose collection $\mathbf{C}^{unc} \subset \mathbf{C}^{all}$ such that:

$$\forall c \in \mathbf{C}^{unc}[\text{MoU}_{\Theta}(c) = \text{True}] \tag{5.6}$$

- Partition $\mathbf{C}^{unc}$ to form clusters $\mathcal{L} \subset \mathcal{P}(\mathbf{C}^{unc})$ such that:

$$\forall \mathbf{C} \in \mathcal{L}[\forall c_1, c_2 \in \mathbf{C}[f_l(c_1) = f_l(c_2)]] \tag{5.7}$$

where f_l is the learning feature function

- Create annotation set $\mathcal{L}^a$, consisting of the n largest clusters from $\mathcal{L}$ under the constraint:

$$\forall \mathbf{C}_1, \mathbf{C}_2 \in \mathcal{L}^a[\text{repr}(\mathbf{C}_1) \neq \text{repr}(\mathbf{C}_2)] \tag{5.8}$$

where repr($\mathbf{C}$) is the most frequent consituent in $\mathbf{C}$

- Present clusters $\mathcal{L}^a$ as samples to the annotator.
- Reestimate training distribution parameters $\Longrightarrow \Theta$.

Figure 5.5: Learning process

phrases are either sensitive, non-sensitive or indeterminate (explained below). The approach is formalised in Figure 5.5. The subsumption constraint (5.5) ensures that the annotator is not presented with both a nonterminal constituent and one more of its subconstituents concurrently as this can result in wasted annotation. The nonduplicity constraint for representatives (5.8) serves to diversify the samples presented for annotation, once again to avoid wasted annotation. Under the designated sample selection constraints, our learning strategy takes account of the three measures specified by (Shen et al. 2004), *informativeness*, *representativeness* and *diversity*, in a computationally efficient manner.

If a sample is deemed *indeterminate* by the annotator its related constituent probabilities are left unchanged. An indeterminate sample implies that its form and context are insufficient to distinguish its sensitivity. One technique to resolve such cases is to use *adaptive features* to enrich the representation of the internal and/or external structure for constituents related to the sample, an avenue of research to be investigated in a future study.

As in (Nigam, McCallum, Thrun & Mitchell 2000), the unlabeled samples are probabilistically labeled using the current model parameters (the E-step) and then the model parameters are re-estimated by incorporating the unlabeled sample estimates (the M-step).

Initial Estimates

Corpus analysis reveals that the class of sensitive words is relatively small. However, given some linguistic and orthographic foreknowledge, it is clear that for certain classes

f_1^i (internal)	f_1^c (contextual)	f_2 (lexical)	f_l (learning)
orthography top structure	parent constituent label inner left const. label 2nd inner left const. label inner right const. label	surface form part-of-speech	f_1^i + f_1^c + outer left const. label outer left const. token

Figure 5.6: Feature selection

Uncertain	Non-sensitive
$P(S\|c) = 0.5$	$P(S\|c) = 0.1$
proper names, titles u.c. single letters, nouns, lexical verbs, adjectives numbers/mixed alphanum	all other tokens

Figure 5.7: Prior estimation

of words the prior probability of sensitivity is significantly higher than the overall class distribution size would suggest. We can therefore posit a simple initial distribution over the class posterior estimates for the unlabeled samples such that the learner will be guided away from the majority of samples that are trivially non-sensitive, and toward samples we are less certain about. This is implemented by assigning terminal constituents to one of two sets, the first representing classes of terms for which there is some measure of uncertainty as to their sensitivity and the second those that are likely to be non-sensitive. Initial class posteriors for terms in the first set are balanced, while those in the second set are weighted in favour of nonsensitivity (Figure 5.7).

Initial nonterminal constituent estimates are made on the basis of their current subconstituent estimates by expression (5.3). The initial estimates for all constituents are gradually superceded by more accurate estimates as learning progresses.

Feature Selection

Features are chosen to characterise a constituent on the basis of its general orthographic and linguistic properties (f_1) as well as on its specific lexical form (f_2). For backoff purposes, we partition f_1 into two subfunctions, f_1^i and f_1^c, representing a constituent's general internal and contextual properties respectively.

After experimenting with a variety of features, we settled on the configurations shown in Figure 5.6. *Orthography* consists of nine non-mutually-exclusive properties: *first letter capitalised*, *all capitalised*, *solo capital*, *integer-like number* (plus length), *float-like number*, *contains non-alphanumeric character*, *contains period*, *contains hyphen* and *contains case or alphanumeric mix*.

Figure 5.2 gives an example of the features calculated for a given focus constituent. All features are domain independent, capturing only the general syntactic and orthographic form of the data.

Backoff

To handle sparsity, a simple backoff strategy is used such that if a terminal constituent's specific lexical form is unrecognised, its posterior is estimated based only on its more general orthographic and syntactic features. Similarly, if a constituent's context is unseen its estimate is based solely on its internal features. This approach is formalised as:

$$P(y|c) = \begin{cases} \lambda P(y|f_1(c)) + (1\text{–}\lambda)P(y|f_2(c)) & \text{no unseen} \\ \lambda P(y|f_1^i(c)) + (1\text{–}\lambda)P(y|f_2(c)) & f_1^c \text{ unseen} \\ P(y|f_1(c)) & f_2\text{unseen} \\ P(y|f_1^i(c)) & f_2\text{,}f_1^c \text{ unseen} \\ \text{Initial estimate} & \text{all unseen} \end{cases}$$

Post Processing

Once an instance of a constituent type has been classified as sensitive, we classify all other tokens of this type in the document as sensitive. In addition, we conjoin the elements of contiguous types classified as sensitive and classify further occurrences of these conjoined elements as sensitive in order to make the approach robust to incorrect or irrelevant syntactic boundaries inserted by the parser.

5.6 Experimental Method

We now discuss issues relating to the methodology we use in our experiments, such as evaluation and input cost quantification. In the following, we will refer to the Lingpipe HMM-based classifier as 'LP', and the alternative model presented in this paper as 'PCM', standing for Probabilistic Constituent Model.

5.6.1 Evaluation Measures

Evaluating the anonymisation task raises issues similar to those faced in NER evaluation. Complications arise due to the comparison of boundaries and partial matches. Arguably the simplest strategy is to evaluate the sensitivity of each token on an individual basis, with recall, precision and F_1 defined in the usual manner:

$$\mathrm{r} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \qquad \mathrm{p} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \qquad \mathrm{F}_1 = \frac{2\mathrm{pr}}{\mathrm{p} + \mathrm{r}}$$

where

TP = count of sensitive tokens correctly identified
FN = count of sensitive tokens missed
FP = count of non-sensitive tokens spuriously identified as sensitive

In one sense, this is a well-motivated approach, bearing in mind that a partially-anonymised reference is increasingly hard to identify as more of its constituent terms are anonymised, eg. *Lee* . . . . . . is preferable to *Lee* . . . *Oswald*.

However, the anonymisation task is actually defined in terms of discrete references, not individual tokens, so arguably it is better to evaluate each referential span as a single item. This raises the question of what to do if a reference is only partially identified (eg. *Smith* instead of *Will Smith*) or if the boundaries are too wide, or crossing (eg. *Jack 's house* instead of just *Jack*).

One approach that attempts to take some of these complications into account is the scheme specified in the MUC guidelines for evaluating information extraction tasks where multiple word spans can represent single items of information.[4]

Another popular approach is that used by the CoNLL community for the NER task, where no credit is given unless an entity reference is fully and correctly identified. In this scheme a partially identified reference is counted both as a false negative and positive. Consequently, overall scores tend to be significantly lower than in either the token-level or MUC evaluation schemes.[5]

To facilitate as broad a range of comparison as possible, we report recall, precision and F_1 under all three evaluation schemes. We have developed our own evaluation scripts for the token-based and MUC schemes and use the CoNLL evaluation script available from the CoNLL website.

5.6.2 Syntactic Analysis

As before, RASP (1.3) is used to generate syntax trees for the PCM classifier. We run the parser largely "as is", the only significant domain-related adaptation being the addition of a number of interjectory terms commonly used in conversational language to the lexicon (*Cheers*, *Hey*, *Regards*, etc.).

5.6.3 Experimental Procedure

We carry out two experimental phases, with their respective aims:

1. To compare the two classification models presented: LP and PCM.
2. To measure anonymisation performance as a function of the quantity of human input required in the training process.

All annotation was carried out by the author, and we quantify the cost of human input by measuring the amount of time spent annotating between each training iteration. For the first set of experiments, we use the *Development* set to compare supervised performance of the LP and PCM classifiers. In the second set, we use the *Training* set to demonstrate that with the iterative learning strategy presented in this study, better accuracy can be achieved by utilising more data without a concomitant increase in annotation cost.

5.7 Results and Analysis

5.7.1 Results

Table 5.1 presents the results for the comparison of LP and PCM when trained on the 472 sentences of the *Development* set. PCM is clearly superior to LP on both the *blanket*

[4] *www.itl.nist.gov/iaui/894.02/related_projects/muc*

[5] *www.cnts.ua.ac.be/conll*

		Blanket			*Selective*		
Eval	*Model*	Rec (%)	Prec (%)	F_1 (%)	Rec (%)	Prec (%)	F_1 (%)
TOK	LP	71.15	77.92	74.38	62.57	66.79	64.61
	PCM	83.71	78.99	81.28	71.15	67.36	69.20
MUC	LP	66.05	76.08	70.71	56.11	62.43	59.10
	PCM	82.64	74.50	78.35	71.44	61.99	66.38
CoNLL	LP	61.38	69.84	65.34	50.23	55.67	52.81
	PCM	76.34	68.04	71.95	63.88	55.74	59.97

Table 5.1: Anonymisation results (models trained on the *Development* set)

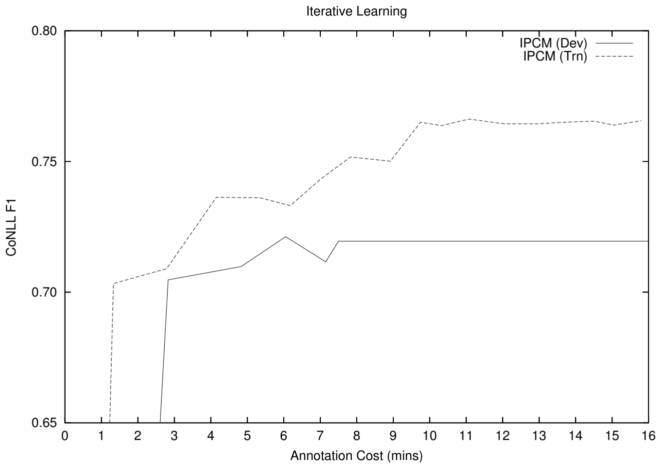


Figure 5.8: PCM (Dev. vs Trn. Set)

and *selective* anonymisation tasks and we take this as indication that the syntactic nature of the PCM renders it better able to generalise in the presence of limited training data by utilising generic syntactic contexts that are indicative of sensitive references. Both techniques fare worse on the *selective* variant of the task, which is harder due to the fact that there are often only contextual distinctions between sensitive and non-sensitive references. For instance, a place name appearing as part of an address (eg. *Bond Street, London*) is considered sensitive, whereas the same term occurring in free text, eg. *I was in London last Friday*, is considered non-sensitive. In the blanket case, it would be considered sensitive in both these contexts. Instances such as these must be differentiated via their context, which worsens the sparsity problem, especially in the presence of limited training data.

Figure 5.8 plots the performance of the PCM when trained iteratively as a function of the cost of annotation. (Dev) denotes iterative training on the small *Development* set while (Trn) denotes use of the larger *Training* set. The graph demonstrates that using a larger dataset results in improved performance without incurring greater annotation costs. In the PCM training procedure, annotation decisions are made about whole clusters, and the bigger the training set, the larger these clusters are likely to be. Consequently, in general, more samples will be labeled when using a larger training set without requiring more annotation decisions.

After around 8 mins annotation, the training procedure converges for the *Develop-*

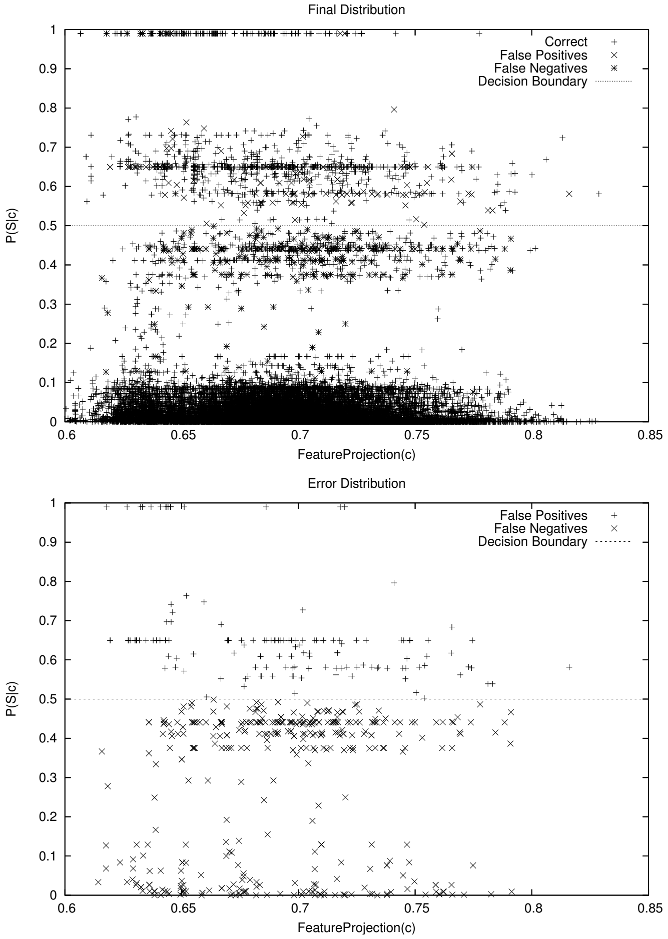


Figure 5.9: Final (top) and error (bottom) distributions

ment set, as all samples fall outside the margin of uncertainty. For the *Training* set, performance levels off after around 10 mins annotation. Though we would expect the rate of performance improvement to diminish over time, it would be useful to know to what extent the error could be reduced by further annotation. In the next section we seek to provide at least a speculative answer to that question.

5.7.2 Discussion

Anonymisation is a complex issue. Any distortion of the data and ensuing loss of information is likely to have some impact on the usefulness of a given dataset, and in each case a decision must be made as to whether any form of anonymisation is feasible. For instance, removing brand-names from email text for spam filtering research might be considered an unacceptable distortion of the nature of unsolicited email, and could thus be seen to jeopardise the validity of research into the problem.

Bearing in mind the current state of NLP technology, it is clear that automatic textual anonymisation must realistically be viewed as an aid to, rather than a replacement for manual anonymisation of sensitive data. An automatic procedure cannot guarantee 100% reliability, even if the parameters of the task can be clearly defined (which is not always the case for anonymisation), and some form of manual checking will need to be carried out to validate the results of the procedure, most importantly to neutralise sensitive references that have evaded detection.

If a probabilistic model is employed (either native or derived) it would be helpful if the final model parameters could be used to point the validator toward uncertain instances, as these represent the boundary cases where misidentification is most likely to have occurred. It would then be up to the validator to decide whether or not he/she can 'safely' ignore instances lying further from the decision boundary. In light of this, when evaluating a probabilistic anonymisation procedure it would be informative to know what percentage of misidentified instances lie near the decision boundary, and also the concentration of misidentified instances in this area (for in the limit *all* remaining instances might be located near the decision boundary, in which case such information is meaningless to the validator). In reality an approach in which misidentified instances occur in high concentration around the decision boundary is likely to be more useful than an approach that achieves greater accuracy but cannot reliably point the validator toward potential misidentifications.

Figure 5.9 shows the terminal constituent posterior distribution, first for the entire test set and then just for the misclassified instances, when using the trained PCM (*Trn*) model (after 16 mins annotation). By examining this distribution it can be observed that approximately 5% of all instances lie within a 0.2 probability margin either side of the decision boundary, and approximately 66% of the misclassified instances lie within the same margin. Thus, by manually examining the 5% of instances lying within this margin, the validator can expect to reduce the total number of errors by around two-thirds.

Approximately 30% of misclassified instances lie at relative extremes of the posterior distribution, and are thus unlikely to be classified correctly irrespective of further training. This constitutes an optimistic upper bound on accuracy of around 0.95 F_1 (token). If we estimate that approximately half of the remaining uncertain instances would be correctly classified given enough training, we arrive at a more realistic upper bound of around 0.90 F_1. Unfortunately we cannot reliably establish how much annotation would actually be required to achieve this level of performance. We can, however, infer from Figure 5.8 that the true upper bound would probably be approached slowly.

5.7.3 Error Analysis

As might be expected, most errors are caused by terms that are orthographically misleading. Some commonly problematic instances include:

- Complex, sensitive references containing many commonly non-sensitive terms, eg. *the Royal Society for the Protection of Birds*. Some of the terms in this reference are usually non-sensitive(*for*, *the*) and capitalisation of common nouns in the email domain (*Protection*, *Birds*) is not particularly suggestive of sensitivity as it is often used simply for emphasis (*Get the New Version of Messenger!*).

- Uncapitalised sensitive terms that look like common non-sensitive terms, of which there are numerous instances in informal text (*penny*, *windows*, *west road*)

- Capitalised references to non-sensitive entities (*New Year*, *God*, *Hemisphere*)

- Non-sensitive references and turns of phrase that do not refer to real world entities yet are functionally and orthographically indistinct. (*the Bogey Man*, *Bill No Mates*)

5.8 Conclusions and Future Work

We have presented a formal description of the automatic textual anonymisation problem and considered how the characteristics of the task affect the manner in which it is approached. We have presented a new corpus of personal email text as a benchmark for evaluating and comparing anonymisation techniques within this domain, *pseudonymised* to allow public release. We have presented a syntactically-motivated probabilistic approach to the anonymsation problem which uses an iterative clustering strategy to learn model parameters under the active learning paradigm. Finally we have reported initial results for the task using *Lingpipe*, an off-the-shelf state-of-the-art HMM-based classifier and the alternative model presented in this study. Our aim is to raise awareness of the issue of anonymisation and encourage further research into methods for approaching the task and discussion into alternative perspectives on the nature of the problem.

Avenues for future research include:

- Application of improved models to the anonymisation task, eg. conditional random fields (CRFs) incorporating syntactic features.

- Investigation of anonymisation in different domains, such as medical record data.

- Investigation into alternative forms of the problem, such as authorship anonymisation through language adaptation.

Chapter 6

Hedge Classification

In this chapter we investigate automatic classification of speculative language, or 'hedging', in scientific literature from the biomedical domain. This is a relatively new topic within NLP and this is the first study to focus on the hedge classification task from a machine learning perspective. Our contributions include a precise description of the task including annotation guidelines, theoretical analysis and discussion. We argue for separation of the acquisition and classification phases in semi-supervised machine learning, and present a probabilistic acquisition model which is evaluated both theoretically and experimentally. We explore the impact of different sample representations on classification accuracy across the learning curve and demonstrate the effectiveness of using machine learning for the hedge classification task. Finally, we examine the weaknesses of our approach and point toward avenues for future research. The work presented in this chapter is an extended version of work published in the proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, *ACL 2007* (Medlock & Briscoe 2007) and also under review for the *Journal of Biomedical Informatics.*

6.1 Introduction

The automatic processing of scientific papers using NLP and machine learning (ML) techniques is an increasingly important aspect of technical informatics. In the quest for a deeper machine-driven 'understanding' of the mass of scientific literature, a frequently occurring linguistic phenomenon that must be accounted for is the use of *hedging* to denote propositions of a speculative nature. As an example, consider the information conveyed by each of the following examples:

1. *Our results prove that XfK89 inhibits Felin-9.*
2. *Our results suggest that XfK89 might inhibit Felin-9.*

The second example contains a hedge, signaled by the use of *suggest* and *might*, which renders the proposition *inhibit(XfK89→Felin-9)* speculative.

For an example of why analysis of hedging is important for automatic text processing, consider a system designed to identify and extract interactions between genetic entities in the biomedical domain. Case 1 above provides clear textual evidence of such an interaction and justifies extraction of *inhibit(XfK89→Felin-9)*, whereas case 2 provides only weak evidence for such an interaction.

Hedging occurs across the entire spectrum of scientific literature, though it is particularly common in the experimental natural sciences. In this study we consider the problem of learning to automatically classify sentences containing instances of hedging, given only a very limited amount of annotator-labeled 'seed' data. This falls within the *semi-supervised* ML framework, for which a range of techniques have been previously explored. The contributions of our work are as follows:

1. We provide a clear description of the problem of hedge classification and offer an improved and expanded set of annotation guidelines, along with illustrative examples, which as we demonstrate experimentally are sufficient to induce a high level of agreement between independent annotators.
2. We discuss the specificities of hedge classification as a semi-supervised ML task.
3. We argue for the separation of the acquisition and classification phases in semi-supervised learning.
4. We derive a probabilistic acquisition model and use it to motivate our approach.
5. We analyze our learning model both theoretically and experimentally, reporting promising results for the task on a new publicly-available full-text dataset.[1]

6.2 Related Work

6.2.1 Hedge Classification

While there is a certain amount of literature within the linguistics community on the use of hedging in scientific text, eg. (Hyland 1994), there is little of direct relevance to the task of classifying speculative language from an NLP/ML perspective.

The most clearly relevant study is Light, Qiu & Srinivasan (2004). They introduce the problem using examples drawn from the biomedical domain, and address the question of whether there is sufficient agreement among humans about what constitutes a speculative assertion to make the task viable from a computational perspective. At first they attempt to distinguish between two shades of speculation: strong and weak, but fail to garner sufficient agreement for such a distinction to be reliably annotated. However, they conclude that it is feasible to draw a reliable distinction between speculative and non-speculative sentences. They focus on introducing the problem, exploring annotation issues and outlining potential applications rather than on the specificities of the ML approach, though they do present some results using a manually crafted substring matching classifier and a supervised SVM on a collection of *Medline* abstracts. We will draw on this work throughout our presentation of the task.

Mercer & Marco (2004) perform some analysis of hedging in the context of citation function, though they do not directly consider the task of hedge classification.

6.2.2 Semi-Supervised Learning

Recent years have witnessed a significant growth of research into semi-supervised ML techniques for NLP applications. Different approaches are often characterised as either *multi-* or *single-view*, where the former generate multiple 'views' on the data and perform mutual

[1] available from *www.cl.cam.ac.uk/ bwm23/*

bootstrapping. This idea was formalised by Blum & Mitchell (1998) in their presentation of *co-training* which they show to be a powerful approach given the assumptions that: 1) each view is *sufficient* for classification, and 2) the views are conditionally independent given the class label. These assumptions very rarely hold in real data, but co-training can still be effective under related but weaker conditions (Abney 2002). Co-training has also been used for named entity recognition (NER) (Collins & Singer 1999), coreference resolution (Ng & Cardie 2003), text categorization (Nigam et al. 2000) and improving gene name data (Wellner 2005). A number of researchers have proposed variants on the co-training idea. For example, rather than partitioning the feature space, Goldman & Zhou (2000) generate multiple views by utilising two different machine learners, each of which is then used to bootstrap the other.

Conversely, single-view learning models operate without an explicit partition of the feature space. Perhaps the most well known of such approaches is *expectation maximization* (EM), used by Nigam et al. (2000) in the context of learning from a combination of labeled and unlabeled data for text categorization. Others have proposed variations on the basic EM algorithm, for instance Ng & Cardie (2003) present a two-tiered bootstrapping approach (EM-FS) in which EM is combined with a feature selection procedure to enhance its performance.

Another single-view algorithm occuring in the literature is called *self-training*, in which a labeled pool is incrementally enlarged with unlabeled samples for which the learner is most confident. Early work by Yarowsky (1995) on WSD (word sense disambiguation) falls within this framework. He proposed a bootstrapping algorithm for learning new patterns given existing ones in an iterative process, utilising the redundancy inherent in the fact that the sense of a word is constrained by its current discourse usage (one sense per discourse), and also by local contextual cues. Banko & Brill (2001) use 'bagging' and agreement to measure confidence on unlabeled samples, and more recently McClosky, Charniak & Johnson (2006) use self-training for improving parse reranking.

Other relevant recent work includes (Zhang 2004), in which random feature projection and a committee of SVM classifiers are used in a hybrid co/self-training strategy for semi-supervised relation classification and (Chen, Ji, Tan & Niu 2006) where a graph based algorithm called *label propagation* is employed to perform semi-supervised relation extraction.

6.3 The Task

Given a collection of sentences, $\mathcal{S}$, the task is to label each sentence as either speculative or non-speculative (*spec* or *nspec* henceforth). Specifically, $\mathcal{S}$ is to be partitioned into two disjoint sets, one representing sentences that contain some form of hedging, and the other representing sentences that do not.

It should be noted that by nature of the task definition, a speculative sentence may contain an arbitrary number of non-speculative assertions, leading to the question of whether hedge classification should be carried out at the granularity of assertions rather than sentences. While there is a strong argument in favour of this approach, it requires the identification of assertion boundaries and thus adds an extra level of complexity to all aspects of the task, from annotation to evaluation. In fact, even if the end goal is to label assertions, sentence level hedge classification can be viewed as an initial stage, after which potentially speculative sentences can be further examined to identify speculative constituents.

In an effort to further elucidate the nature of the task and to aid annotation, we have developed a new set of guidelines, building on the work of Light et al. (2004). It is important to note that at least on a conceptual level, speculative assertions are not to be identified on the basis of the presence of certain designated hedge terms, rather the assessment is made based on a judgement of the author's intended meaning, as revealed by the text.

We begin with the hedge definition given by Light et al. (item 1) and introduce a set of further guidelines to help illucidate various 'grey areas' and tighten the task specification. The following ARE considered instances of hedging:

1. Any assertion relating to a result that does not necessarily follow from the work presented, but could be extrapolated from it (Light et al. 2004), eg:

   This unusual substrate specificity may explain why Dronc is resistant to inhibition by the pan-caspase inhibitor

   Indeed, most mitochondria released all their cytochrome c, suggesting that an enzymatic transport mechanism is probably not involved

   Our results provide the first direct evidence linking RAG1 and RSSs to a specific superfamily of DNA transposons and indicate that the V(D)J machinery evolved from transposons

   A reduction of coverage could be the result of a reduction in dendrite outgrowth

   Thus, nervy likely regulates multiple aspects of neuronal differentiation

2. Relay of hedge made in previous work, eg:

   Dl and Ser have been proposed to act redundantly in the sensory bristle lineage

3. Statements of knowledge paucity, eg:

   How endocytosis of Dl leads to the activation of N remains to be elucidated

   Biochemical analysis of the ubiquitination events regulated by D-mib will be needed to further define the mechanism by which D-mib regulates the endocytosis of Ser in vivo

   There is no clear evidence for cytochrome c release during apoptosis in C. elegans or Drosophila

   There is no apparent need for cytochrome c release in C. elegans, since CED-4 does not require it to activate CED-3

4. Speculative questioning, eg:

   A second important question is whether the roX genes have the same, overlapping or complementing functions

5. Statement of speculative hypothesis, eg:

   To test whether the reported sea urchin sequences represent a true RAG1-like match, we cut off the ring finger motif and repeated the BLASTP search against all GenBank proteins

6. Anaphoric hedge, eg:

   This hypothesis is supported by our finding that both pupariation rate and survival of ...

   The rescue of the D-mib mutant phenotype by ectopic expression of Neur strongly supports this interpretation

The following are NOT considered instances of hedging:

1. Indication of experimentally observed non-universal behaviour, eg:

   proteins with single BIR domains can also have functions in cell cycle regulation and cytokinesis

   These results demonstrate that ADGF-A overexpression can partially rescue the effects of constitutively active Toll signaling in larvae

   IAPs contain at least one BIR domain, and often a carboxy-terminal RING domain

2. Confident assertion based on external work, eg:

   Two distinct E3 ubiquitin ligases have been shown to regulate Dl signaling in Drosophila melanogaster

3. Statement of existence of proposed alternatives, eg:

   Different models have been proposed to explain how endocytosis of the ligand, which removes the ligand from the cell surface, results in N receptor activation

4. Confirmation of previous speculation, eg:

   Here we show that the hemocytes (blood cells) are the main regulator of adenosine in the Drosophila larva, as was speculated previously for mammals

5. Confirmation of already firmly-stated conclusion

   This conclusion is further supported by the even more efficient rescue achieved by ...

6. Negation of previous hedge

   Although the adgf-a mutation leads to larval or pupal death, we have shown that this is not due to the adenosine or deoxyadenosine simply blocking cellular proliferation or survival, as the experiments in vitro would suggest

	F_1^{rel}	κ
Original	0.8293	0.9336
Corrected	0.9652	0.9848

Table 6.1: Agreement Scores

6.4 Data

For our experiments, we used an archive of 5579 full-text papers from the functional genomics literature relating to Drosophila melanogaster (the fruit fly). The papers were converted to XML and linguistically processed using the RASP toolkit (1.3). We annotated six of the papers to form a test set with a total of 380 *spec* sentences and 1157 *nspec* sentences, and randomly selected 300,000 sentences from the remaining papers as training data for the semi-supervised learner. The unlabeled sentences were chosen under the constraints that they must be at least 10 words long and contain a main verb.

6.5 Annotation and Agreement

Two separate annotators were commissioned to label the sentences in the test set, the author and a domain expert with no prior input into the guideline development process. The two annotators labelled the data independently using the guidelines outlined in section 6.3. Relative F_1 (F_1^{rel}) and *Cohen's Kappa* (κ) were then used to quantify the level of agreement. The idea behind relative F_1 (Hripcsak & Rothschild 2004) is to use one of the annotations as the 'gold standard' and compute F_1 for the other. Only the F_1 score for the *spec* class is used, and this value is symmetric, i.e. either annotator can be taken as the gold standard. Cohen's κ measures the agreement between two annotators, corrected for chance, and is defined as:

$$\kappa = \frac{P(a) - P(e)}{1 - P(e)}$$

where $P(a)$ is the relative observed agreement among raters, and $P(e)$ is the probability that agreement is due to chance. We refer the reader to (Artstein & Poesio 2005) for more detailed formulation and further discussion of κ.

The two metrics are based on different assumptions about the nature of the annotation task. F_1^{rel} is founded on the premise that the task is to recognise and label *spec* sentences from within a background population, and does not explicitly model agreement on *nspec* instances. It ranges from 0 (no agreement) to 1 (no disagreement). Conversely, κ gives explicit credit for agreement on both *spec* and *nspec* instances. The observed agreement is then corrected for 'chance agreement', yielding a metric that ranges between -1 and 1. Given our definition of hedge classification and assessing the manner in which the annotation was carried out, we suggest that the founding assumption of F_1^{rel} fits the nature of the task better than that of κ.

Following initial agreement calculation, the instances of disagreement were examined. It turned out that the large majority of cases of disagreement were due to negligence on behalf of one or other of the annotators (i.e. cases of clear hedging that were missed), and that the cases of genuine disagreement were actually quite rare. New labelings were then created with the negligent disagreements corrected, resulting in significantly higher agreement scores. Values for the original and negligence-corrected labelings are reported in Table 6.1.

Annotator conferral violates the fundamental assumption of annotator independence, and so the latter agreement scores do not represent the true level of agreement; however, it is reasonable to conclude that the actual agreement is approximately lower bounded by the initial values and upper bounded by the latter values. In fact even the lower bound is well within the range usually accepted as representing 'good' agreement, and thus we are confident in accepting human labeling as a gold-standard for the hedge classification task. For our experiments, we use the labeling of the genetics expert, corrected for negligent instances.

6.6 Discussion

In this section we provide some justification as to why we expect a hedge classifier to be learnable in a semi-supervised manner.

The acquisition of new information about a particular target function from unlabeled data depends upon the existence of redundancy in the specification of the target function, even if the feature space cannot be explicitly partitioned into conditionally independent sets. This idea can be formalised as follows: given a particular feature, whose presence we will denote by f_1, a target function Y and a learned hypothesis H, let us suppose that f_1 is a good indicator of a certain target function value y, eg. $P(Y=y|f_1) \approx 1$. We will also assume that this is known to the learner, eg. $P(H=y|f_1) = P(Y=y|f_1)$. To infer new information about Y using an unlabeled source, there must exist some feature f_2, also a good indicator of $Y=y$, such that the following conditions hold:

1. $P(f_2|f_1) > P(f_2)$
2. $P(f_2|f_1, Y=y) < 1$

Condition 1 states that the features must not be negatively correlated, i.e. it must be possible to infer from instances containing f_1 that f_2 is also a good indicator of $Y=y$, while condition 2 states that the positive correlation between the two features, conditioned on the target class, must not be too tight, otherwise the learning process will grind to a halt. We can generalise from single features to 'rules' or 'views' that combine multiple features, but the same principles apply. Taken together, these conditions are a less precise, but for our task more intuitive version of the *weak rule dependence* condition of (Abney 2002), which is itself a relaxed version of the *conditional independence assumption* of (Blum & Mitchell 1998).

Analysis of the hedge classification task reveals potential redundancy of the above form that should be exploitable by a suitably chosen semi-supervised learner. We begin by assuming that features in our model are single terms, based on the intuition that many hedge cues are single terms (*suggest*, *likely* etc.) and due to the success of 'bag of words' representations in many learning tasks to date. Later, we will consider possible techniques for enriching the representation.

Consider again the example speculative sentence from earlier: *"These results suggest that XfK89 might inhibit Felin-9."* Both *suggest* and *might* are hedge cues, and it is plausible to assume that they also occur within speculative sentences in other contexts, for instance *"We suspect there might be an interaction between XfK89 and Felin-9."*. Now, for $f_1 = suggest$ and $f_2 = might$, we can examine the conditions specified above:

1. $P(might|suggest) > P(might)$
2. $P(might|suggest, Y = spec) < 1$

The required values can be estimated from our data, yielding the following:

$$\begin{aligned} P(might|suggest) &= 0.037 \\ P(might) &= 0.012 \end{aligned}$$

Given the (quite reasonable) approximation that *suggest* is always used as a hedge cue, $P(might|suggest) = P(might|suggest, Y = spec)$ and both conditions hold.

While such evidence suggests that the task is feasible, there are a number of factors that make our formulation of hedge classification both interesting and challenging from a semi-supervised learning perspective. Firstly, each sample contains a potentially large number of irrelevant features, as hedging modifies the certainty with which an assertion is made, but in general does not modify the assertion itself, rendering most of the actual content of an assertion irrelevant. However, hedge cues come from a wide range of linguistic categories, mitigating against techniques such as traditional stopword filtering, and take many different forms. Consequently there are no obvious ways of removing irrelevant features without also losing potential hedge cues. Exacerbating this problem is the fact that speculative sentences may contain many non-speculative assertions, each potentially adding a large number of irrelevant features.

Such characteristics are in contrast to much previous work on semi-supervised learning, where for instance in the case of text categorization (Blum & Mitchell 1998, Nigam et al. 2000) almost all content terms are to some degree relevant, and irrelevant features are usually stopwords and can easily be filtered out. In the same vein, for the case of entity/relation extraction and classification (Collins & Singer 1999, Zhang 2004, Chen et al. 2006) the context of the entity or entities in consideration provides a highly relevant feature space, and such studies are often set up such that only entities which fulfill some contextual criteria are considered (Collins & Singer 1999).

Another interesting factor in our formulation of hedge classification that sets it apart from previous work is that the class of non-speculative sentences is defined on the basis of the *absence* of hedging, rather than on any positive characteristic. This makes it difficult to model the *nspec* class directly, and also hard to select a reliable set of *nspec* seed sentences, as by definition at the beginning of the learning cycle the learner has little knowledge about what a hedge looks like. The *nspec* seed problem is addressed in section 6.10.3.

In this study we will develop a learning model based around the idea of iteratively predicting labels for unlabeled training samples. This is the basic paradigm for both co-training and self-training; however we will generalise by framing the task in terms of the acquisition of labelled training data, from which a supervised classifier can subsequently be learned. It is our contention that there are good reasons for making the distinction between acquiring training data and classification, based on the observation that, while clearly related, the tasks are not the same. This distinction will become clearer in the next section when we develop a formal model for the learning procedure; however, using the arguments put forward in this discussion one can see informally where some of the distinctions lie. As we have seen, redundancy in the representation is crucial for acquiring new training samples; however this is not the case for classification. The aim of a classifier is to learn an accurate mapping between samples and target classes, and this does not require feature redundancy; in fact it is often beneficial to *reduce* redundancy by using features that specify the target classes more precisely. Given this insight, it may be advantageous to use different representations for the acquisition and classification phases, in addition to employing different learning models.

A related, though somewhat orthogonal argument can be made from the point of view of data sparsity. At the start of the acquisition phase, there is only a very limited amount of training data (the seed samples), and a complex representation is likely to suffer excessively from issues of data sparsity. However, once a sufficiently large training set has been induced, this becomes much less of an issue, and a more complex representation might indeed be beneficial.

6.7 A Probabilistic Model for Training Data Acquisition

In this section, we derive a simple probabilistic model for acquiring training data for a given learning task, and use it to motivate our approach to semi-supervised hedge classification.

Given:

- sample space $\mathcal{X}$
- set of target concept classes $\mathcal{Y} = \{y_1 \dots y_n\}$
- target function $Y : \mathcal{X} \rightarrow \mathcal{Y}$
- set of seed samples for each class $\mathcal{S}_1 \dots \mathcal{S}_n$ where $\mathcal{S}_i \subset \mathcal{X}$ and $\forall \mathbf{x} \in \mathcal{S}_i [Y(\mathbf{x}) = y_i]$
- set of unlabeled samples $\mathcal{U} = \{\mathbf{x}_1 \dots \mathbf{x}_K\}$

Aim: *Infer a set of training samples $\mathcal{T}_i$ for each concept class y_i such that* $\forall \mathbf{x} \in \mathcal{T}_i [Y(\mathbf{x}) = y_i]$

Now, it follows that $\forall \mathbf{x} \in \mathcal{T}_i [Y(\mathbf{x}) = y_i]$ is satisfied in the case that $\forall \mathbf{x} \in \mathcal{T}_i [P(y_i|x) = 1]$, which leads to a model in which $\mathcal{T}_i$ is initialised to $\mathcal{S}_i$ and then iteratively augmented with the unlabeled sample(s) for which the posterior probability of class membership is maximal. Formally:

At each iteration:

$$\begin{aligned} &\mathcal{T}_i \leftarrow \mathbf{x}_j (\in \mathcal{U}) \\ &\qquad \text{where } j = \arg\max_j [P(y_i|\mathbf{x}_j)] \end{aligned} \tag{6.1}$$

Expansion with Bayes' Rule yields:

$$\begin{aligned} & \arg\max_j \left[P(y_i|\mathbf{x}_j) \right] \\ = \; & \arg\max_j \left[\frac{P(\mathbf{x}_j|y_i) \cdot P(y_i)}{P(\mathbf{x}_j)} \right] \end{aligned} \tag{6.2}$$

An interesting observation is the importance of the sample prior $P(\mathbf{x}_j)$ in the denominator, often ignored for classification purposes because of its invariance to class. We can expand further by marginalising over the classes in the denominator (equation 6.2) and rearranging, yielding:

$$\arg\max_j \left[\frac{P(\mathbf{x}_j|y_i) \cdot P(y_i)}{\sum_{n=1}^{N} P(y_n)P(\mathbf{x}_j|y_n)} \right] \tag{6.3}$$

so we are left with the class priors and class-conditional likelihoods, which can usually be estimated directly from the data, at least under limited dependence assumptions. The class priors can be estimated based on the relative distribution sizes derived from the current training sets:

$$P(y_i) = \frac{|\mathcal{T}_i|}{\sum_k |\mathcal{T}_k|} \tag{6.4}$$

where $|\mathcal{T}|$ is the number of samples in training set $\mathcal{T}$.

If we assume feature independence, which as we will see for our task is not as gross an approximation as it may at first seem, we can simplify the class-conditional likelihood in the well known manner:

$$P(\mathbf{x}_j|y_i) = \prod_k P(x_{jk}|y_i) \tag{6.5}$$

and then estimate the likelihood for each feature:

$$P(x_k|y_i) = \frac{\alpha P(y_i) + f(x_k, \mathcal{T}_i)}{\alpha P(y_i) + |\mathcal{T}_i|} \tag{6.6}$$

where $f(x, \mathcal{S})$ is the number of samples in training set $\mathcal{S}$ in which feature x is present, and α is a universal smoothing constant, scaled by the class prior. This scaling is motivated by the principle that without knowledge of the true distribution of a particular feature it makes sense to include knowledge of the class distribution in the smoothing mechanism. Smoothing is particularly important in the early stages of the learning process when the amount of training data is severely limited resulting in unreliable frequency estimates.

6.8 Hedge Classification

We will now consider how to apply this learning model to the hedge classification task. As discussed earlier, the speculative/non-speculative distinction hinges on the presence or absence of a few hedge cues within the sentence. Working on this premise, all features are ranked according to their probability of 'hedge cue-ness':

$$P(spec|x_k) = \frac{P(x_k|spec) \cdot P(spec)}{P(spec)P(x_k|spec) + P(nspec)P(x_k|nspec)} \tag{6.7}$$

which can be computed directly using (6.4) and (6.6). The m most probable features are then selected from each sentence to compute (6.5) and the rest are ignored. This has the dual benefit of removing irrelevant features and also reducing dependence between features, as the selected features will often be non-local and thus not too tightly correlated.

Note that this idea differs from traditional feature selection in two important ways:

1. Only features indicative of the *spec* class are retained, or to put it another way, *nspec* class membership is inferred from the absence of strong *spec* features.
2. Feature selection in this context is *not* a preprocessing step. The classes are not re-modelled after selection; rather the original estimates are used. This has the effect of heavily skewing the posterior estimates in favour of the *spec* class, but this is acceptable for ranking purposes.

Of course, this 'one-sided' feature selection technique could be carried out prior to class estimation as a preprocessing step; however, we would not expect this to be effective, as the *nspec* class would then be severely misrepresented, and the *spec* estimates would suffer accordingly. Later we demonstrate this to be the case experimentally (6.10.5).

6.9 Classification

The acquisition procedure returns a labeled data set for each class, from which a classifier can be trained. We use SVM^{light} (1.3), and for comparison purposes, we derive a simple probabilistic classifier using the estimates from our learning model by:

$$\mathbf{x}_j \rightarrow spec \quad \text{if} \quad P(spec|\mathbf{x}_j) > \sigma \tag{6.8}$$

where σ is an arbitrary threshold used to control the precision/recall balance.

6.10 Experimental Evaluation

6.10.1 Method

To examine the practical efficacy of the learning and classification models we have presented, we use the following experimental method:

1. Generate seed training data: $\mathcal{S}_{spec}$ and $\mathcal{S}_{nspec}$
2. Initialise: $\mathcal{T}_{spec} \leftarrow \mathcal{S}_{spec}$ and $\mathcal{T}_{nspec} \leftarrow \mathcal{S}_{nspec}$
3. Iterate:
   - Order $\mathcal{U}$ by $P(spec|\mathbf{x}_j)$ (expression 6.3)
   - $\mathcal{T}_{spec} \leftarrow$ most probable batch
   - $\mathcal{T}_{nspec} \leftarrow$ least probable batch
   - Train classifier using $\mathcal{T}_{spec}$ and $\mathcal{T}_{nspec}$
   - Compute *spec* recall/precision BEP (break-even point) on the test data

The batch size for each iteration is set to $0.001 * |\mathcal{U}|$. After each learning iteration, we compute the precision/recall BEP for the *spec* class using both classifiers trained on the current labeled data. We use BEP because it helps to mitigate against misleading results due to discrepancies in classification threshold placement. Disadvantageously, BEP does not measure a classifier's performance across the whole of the recall/precision spectrum (as can be obtained, for instance, from receiver-operating characteristic (ROC) curves), but for our purposes it provides a clear, abstracted overview of a classifier's accuracy given a particular training set.

Rank	$\alpha = 0$	$\alpha = 1$	$\alpha = 5$	$\alpha = 100$	$\alpha = 500$
1	interactswith	suggest	suggest	suggest	suggest
2	TAFb	likely	likely	likely	likely
3	sexta	may	may	may	may
4	CRYs	might	might	These	These
5	DsRed	seems	seems	results	results
6	Nonautonomous	suggests	Taken	might	that
7	arva	probably	suggests	observations	be
8	inter-homologue	suggesting	probably	Taken	data
9	Mohanty	possibly	Together	findings	it
10	meld	suggested	suggesting	Our	Our
11	aDNA	Taken	possibly	seems	observations
12	Deer	unlikely	suggested	together	role
13	Borel	Together	findings	Together	most
14	substripe	physiology	observations	role	these
15	Failing	modulated	Given	that	together
16	uncommitted	reflecting	unlikely	be	might
17	dist&xAFnct	destruction	These	it	findings
18	descend	cooperative	reflect	strongly	more
19	excretions	Preliminary	results	most	function
20	actinC	outcome	Our	data	is
21	Slit-mediated	insufficient	reflects	mechanism	Taken
22	&x&xB&xB&xBDr	achieve	Rather	important	our
23	lCD	antagonize	together	due	seems
24	VanBerkum	Abd-B	physiology	play	due
25	DE-Cad	inability	modulated	suggests	studies

Table 6.2: Features ranked by $P(spec|x_k)$ for varying α

6.10.2 Parameter Setting

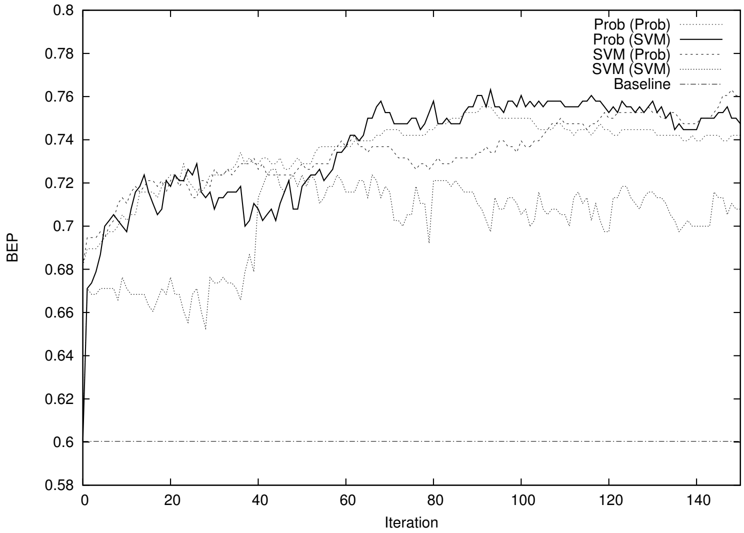
The training and classification models we have presented require the setting of two parameters: the smoothing parameter α and the number of features per sample m. Analysis of the effect of varying α on feature ranking reveals that when $\alpha = 0$, low frequency terms with spurious class correlation dominate and as α increases, high frequency terms become increasingly dominant, eventually smoothing away genuine low-to-mid frequency correlations. This effect is illustrated in Table 6.2, and from this analysis we chose $\alpha = 5$ as an appropriate level of smoothing.

We use $m = 5$ based on the intuition that five is a rough upper bound on the number of hedge cue features likely to occur in any one sentence. We do not contend that these are optimal parameter values for the task, rather that they are sensible.

We use the linear kernel for SVM^{light} with the default setting for the regularization parameter C. We construct binary valued, L$_2$-normalised (unit length) input vectors to represent each sentence, as this resulted in better performance than using frequency-based weights and concords with our presence/absence feature estimates.

6.10.3 Seed Generation

The learning model we have presented requires a set of seeds for each class. To generate seeds for the *spec* class, we extracted all sentences from $\mathcal{U}$ containing either (or both) of the terms *suggest* or *likely*, as these are very good (though not perfect) hedge cues,



Prob (Prob)	denotes our probabilistic learning model and classifier (6.8)
Prob (SVM)	denotes probabilistic learning model with SVM classifier
SVM (Prob)	denotes committee-based model (6.10.4) with probabilistic classifier
SVM (SVM)	denotes committee-based model with SVM classifier
Baseline	denotes substring matching classifier of (Light et al. 2004)

Figure 6.1: Learning curves

yielding 6423 *spec* seeds. Generating seeds for *nspec* is much more difficult, as integrity requires the absence of hedge cues, and this cannot be done automatically. Thus, we used the following procedure to obtain a set of *nspec* seeds:

1. Create initial $\mathcal{S}_{nspec}$ by sampling randomly from $\mathcal{U}$.
2. Manually remove more 'obvious' speculative sentences using pattern matching
3. Iterate:
    - Order $\mathcal{S}_{nspec}$ by $P(spec|\mathbf{x}_j)$ using estimates from $\mathcal{S}_{spec}$ and current $\mathcal{S}_{nspec}$
    - Examine most probable sentences and remove speculative instances

We started with 8830 sentences and after a couple of hours work reduced this down to a (still potentially noisy) *nspec* seed set of 7541 sentences.

6.10.4 Baselines

As a baseline classifier we use the substring matching technique of (Light et al. 2004), which labels a sentence as *spec* if it contains one or more of the following: *suggest*, *potential*, *likely*, *may*, *at least*, *in part*, *possibl*, *further investigation*, *unlikely*, *putative*, *insights*, *point toward*, *promise*, *propose*.

To provide a comparison for our learning model, we implement a more traditional self-training procedure in which at each iteration a committee of five SVMs is trained on randomly generated overlapping subsets of the training data and their cumulative confidence is used to select items for augmenting the labeled training data. For similar work see (Banko & Brill 2001, Zhang 2004).

6.10.5 Initial Results

Figure 6.1 plots accuracy as a function of the training iteration. After 150 iterations, all of the semi-supervised learning models are significantly more accurate than the baseline according to a binomial sign test ($p < 0.01$), though there is clearly still much room for improvement. The baseline classifier achieves a BEP of 0.60 while both classifiers reach approximately 0.76 BEP using our probabilistic acquisition model, with the SVM performing slightly better overall. The weakest combination is the SVM committee-based learning model with an SVM classifier, SVM (SVM). Interestingly though, the probabilistic classifier with the SVM committee-based acquisition model performs competitively with the other approaches. Overall, these results favour a framework in which the acquisition and classification phases are carried out by different models.

6.11 Exploring the Learning Model

Recall from Section 6.7 that in our formulation, training data augmentation proceeds on the basis of sample selection by maximal class posterior:

$$\begin{aligned} &\mathcal{T}_i \leftarrow \mathbf{x}_j (\in \mathcal{U}) \\ &\qquad \text{where } j = \arg\max_j [P(y_i|\mathbf{x}_j)] \end{aligned} \tag{6.9}$$

After expansion:

$$\begin{aligned} & \arg\max_j \,[P(y_i|\mathbf{x}_j)] \\ = \;& \arg\max_j \left[\frac{P(\mathbf{x}_j|y_i)\cdot P(y_i)}{P(\mathbf{x}_j)}\right] \end{aligned} \tag{6.10}$$

Because of its invariance with respect to the sample, the class prior $P(y_i)$ can be eliminated from the numerator, and by taking the log of the result we derive the expression for the *pointwise mutual information (PMI)* between the sample and the class:

$$\begin{aligned} \propto \;& \arg\max_j \left[\frac{P(\mathbf{x}_j|y_i)}{P(\mathbf{x}_j)}\right] \\ \propto \;& \arg\max_j \left[\log\frac{P(\mathbf{x}_j|y_i)}{P(\mathbf{x}_j)}\right] \\ = \;& \arg\max_j \,[\mathrm{PMI}(\mathbf{x}_j, y_i)] \end{aligned} \tag{6.11}$$

This demonstrates that sample selection by maximal class posterior is equivalent to selection by maximal PMI, and raises the question of whether it might be possible to estimate the sample prior $P(\mathbf{x}_j)$ directly from the data, without marginalising. Advantageously, this would allow us (in principle) to rank samples proportionally to $P(\mathit{spec}|\mathbf{x}_j)$ without requiring an estimate of $P(\mathit{nspec}|\mathbf{x}_j)$, and thus avoiding the problematic generation of *nspec* seeds.

Under the feature independence assumption, the sample prior can be factorised into a product of its individual feature priors in a similar manner to the factorisation of the class conditional likelihood (6.5). Rearrangement yields:

$$
\begin{aligned}
& \arg\max_j \left[\log \frac{P(\mathbf{x}_j|y_i)}{P(\mathbf{x}_j)} \right] \\
= \; & \arg\max_j \left[\log \frac{\prod_k P(x_{jk}|y_i)}{\prod_k P(x_{jk})} \right] \\
= \; & \arg\max_j \left[\log \prod_k \frac{P(x_{jk}|y_i)}{P(x_{jk})} \right] \\
= \; & \arg\max_j \left[\sum_k \log \frac{P(x_{jk}|y_i)}{P(x_{jk})} \right] \\
= \; & \arg\max_j \left[\sum_k \mathrm{PMI}(x_{jk}, y_i) \right] \qquad (6.12)
\end{aligned}
$$

We are left with a summation over the PMI values for the individual features within each sample. This calculation is both efficient and highly perspicuous as the contributions of individual features are simply added together, and in fact we can use the per-feature contribution w.r.t the *spec* class, $\log \frac{P(x_k|spec)}{P(x_k)}$, to perform feature selection in the manner discussed earlier (6.8). We already have an expression (6.6) to estimate the per-feature class conditional likelihood:

$$P(x_k|y_i) = \frac{\alpha P(y_i) + f(x_k, \mathcal{T}_i)}{\alpha P(y_i) + |\mathcal{T}_i|} \qquad (6.13)$$

and we use a similar estimate for the feature prior:

$$P(x_k) = \frac{\alpha + f(x_k, \mathcal{T})}{\alpha + |\mathcal{T}|} \qquad (6.14)$$

where $\mathcal{T}$ represents the set containing all the training data, both labeled and unlabeled and $P(y_i) = |\mathcal{T}_i|/|\mathcal{T}|$. In section 6.7 we introduced the idea of prior-scaled smoothing. In the formulation we have presented here it plays a crucial role in the emergence of useful feature-class correlations. Table 6.3 demonstrates this phenomenon. The non-scaled formulation of the per-feature class conditional likelihood is as follows:

$$P(x_k|y_i) = \frac{\alpha + f(x_k, \mathcal{T}_i)}{\alpha + |\mathcal{T}_i|} \qquad (6.15)$$

Few, if any useful correlations emerge when using 6.15; however, when using scaled smoothing, genuine correlations do emerge. The reason for the effectiveness of scaled smoothing is that the amount of smoothing is related to the *current estimate* of the focus class prior in relation to the whole body of training data, which at the start of the learning process is low. This encourages variation in the per-feature class conditional estimate (numerator), while utilising the higher α value to dampen the effect of low frequency terms in the feature prior (denominator). Though many of the terms ranked highly using the scaled estimates appear potentially indicative of speculativity, it is clear that as hedge cues they are not as reliable as the high ranking terms in Table 6.2 based on the marginalised posterior estimates.

	$\alpha = 5$		$\alpha = 100$	
Rank	*Scaled*	*Non-scaled*	*Scaled*	*Non-scaled*
1	suggest	interactswith	suggest	likely
2	likely	Nonautonomous	likely	interactswith
3	Ix	aDNA	Taken	Nonautonomous
4	LRTs	EGFP	Together	aDNA
5	Taken	learns	findings	EGFP
6	Cumulatively	Adelsberger	observations	learns
7	impinges	Ubx&xs	These	Adelsberger
8	&xCopen&xD	polytypic	seems	Ubx&xs
9	FNIII	hairing	Our	polytypic
10	Wingrove	variegation	results	hairing
11	Zalfa	dLglPAR	together	variegation
12	earlystage	t&xBrotein	Altogether	dLglPAR
13	CRN	Dor-dependent	Collectively	t&xBrotein
14	Pfalciparum	icated	Recent	Dor-dependent
15	gel-like	peptidelipid	strongly	icated
16	peroxisomal&xD	&xBlightly	conformational	peptidelipid
17	polyQ-expanded	PRATHER	think	&xBlightly
18	misannotated	Keen	underestimate	PRATHER
19	ratio&xs	C&xB&xBA	most	Keen
20	GENERAL	C̃&xB&xBA	play	C&xB&xBA
21	Miyashiro	&xAFnhibit	seemed	C̃&xB&xBA
22	muscle-identity	mpor&xBn	Given	&xAFnhibit
23	self-recognition	KLARENBERG	rather	mpor&xBn
24	rBm-dNK-&x&xAFs	Stroopper	prove	KLARENBERG
25	Fukui	spersed	roles	Stroopper

Table 6.3: Features ranked by PMI(x_k, *spec*) with and without scaling.

Note that we can easily modify the classifier presented earlier to use the PMI estimates instead of the posterior:

$$\mathbf{x}_j \rightarrow \mathit{spec} \ \ \text{if} \ \ \mathrm{PMI}(\mathbf{x}_j, \mathit{spec}) > \sigma \tag{6.16}$$

A consequence of the new model is that because estimates are being made on the basis of one class and a 'background' corpus, there is not the same motivation for using a binary discriminative classifier.

6.11.1 Experimental Evaluation

The alternative self-training formulation derived above requires a slightly different experimental method:

1. Generate seed training data: $\mathcal{S}_{spec}$
2. Initialise: $\mathcal{T}_{spec} \leftarrow \mathcal{S}_{spec}$
3. Iterate:
    - Order $\mathcal{U}$ by PMI($\mathbf{x}_j$, *spec*) (expression 6.12)
    - $\mathcal{T}_{spec} \leftarrow$ highest ranked batch
    - Train classifier using $\mathcal{T}_{spec}$ and $\mathcal{T}$
    - Compute *spec* recall/precision BEP (break-even point) on the test data

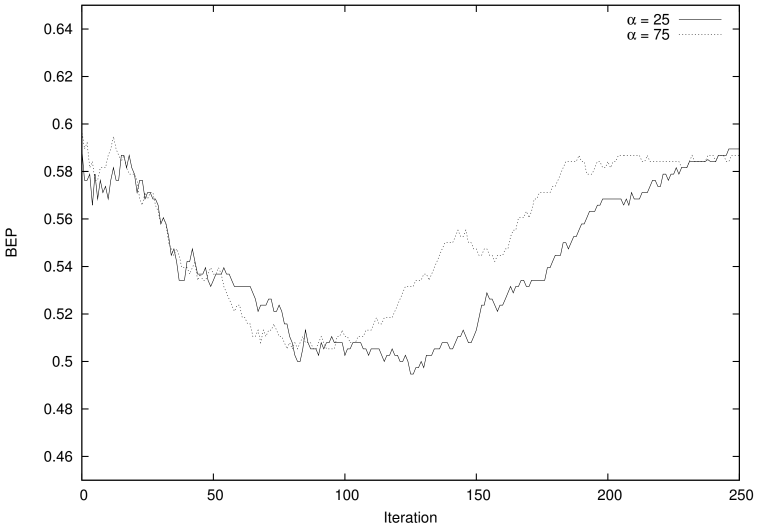


Figure 6.2: Learning curve for PMI acquisition model

We use the classifier defined in 6.16, and the same feature selection technique as in previous experiments (though using PMI instead of the posterior) with $m = 5$.

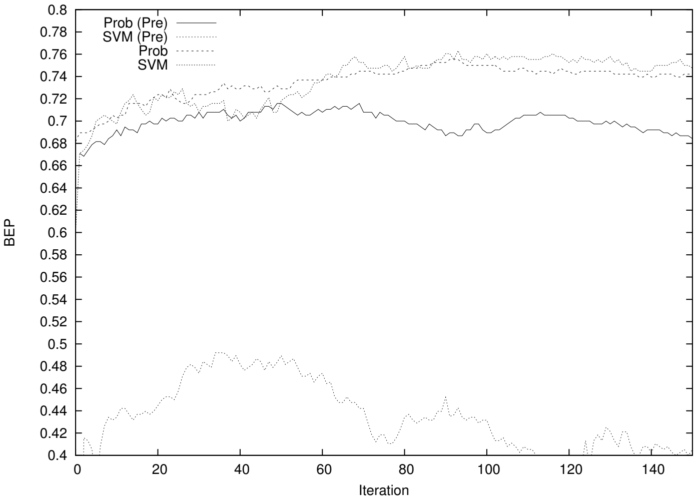
Figure 6.2 plots the learning curve for the alternative one-class PMI model with $\alpha = 25$ and 75. Alternative values of α yielded very similar results, though performance degraded further for lower values, as expected given Table 6.3. Unfortunately, the one-class model is unable to compete with the two-class techniques presented earlier, failing to improve upon the initial classifier learned from the seed data. An explanation for the weakness of the model follows from an examination of its theoretical properties. Samples are chosen on the basis of PMI, given by $\log \frac{P(\mathbf{x}_j|y_i)}{P(\mathbf{x}_j)}$. The weakness here is actually the fact that while our estimate of the sample prior $P(\mathbf{x}_j)$ is quite reliable (given the independence assumption), the class conditional likelihood estimate $P(\mathbf{x}_j|y_i)$ is unreliable at the beginning of the learning cycle as it is estimated only from the seed data. In particular, many genuine feature-class correlations are weak, and even with the scaled smoothing they tend to be 'drowned out' when competing with prior estimates from the entire training corpus. This phenomenon can be observed by comparing the top ranked *spec* features under the two models (Tables 6.2 and 6.3); the former represent quite strong hedge cues, whereas the latter are clearly weaker.

6.12 Further Experiments

We now return to the more successful initial learning model, investigating various aspects of its application and seeking to improve classification accuracy.

6.12.1 Feature Selection

The results presented in Figure 6.1 for the probabilistic classifier use the one-sided feature selection technique outlined in Section 6.8, while the SVM results are obtained without feature selection. Figure 6.3 plots the results of experiments in which we carry out one-



(Pre) denotes feature selection as preprocessing step

Figure 6.3: Learning curves – feature selection as preprocessing step

sided feature selection for both classifiers as a *preprocessing* step, in order to test its expected ineffectiveness when used in the traditional way. As anticipated, both classifiers perform worse in this scenario, with a particularly dramatic decrease in accuracy for the SVM. We attribute this phenomenon to the fact that in the one-sided feature selection scenario, the SVM must discriminate between classes that are both represented by features indicative of the *spec* class; a task at which it is intuitively destined to fail. We also carried out experiments (not reported here) using traditional feature selection on both classes (with χ^2_{max}) for the SVM classifier, and this resulted in poorer performance than using all features.

6.12.2 Seed Refinement

Having gathered initial results for the task, we chose to spend a little more time refining the *nspec* seeds, driven by the hypothesis that even a relatively small number of spurious *spec* sentences amongst the *nspec* seeds could cause the learner to erroneously disregard some of the more subtle *spec* class-feature correlations and thus significantly degrade the diversity of the induced training data.

We again employed the iterative refinement method described above (§6.10.3), and additionally, we used the substring classifier of Light et al. (2004) to extract a list of further potentially speculative sentences from which we removed the genuinely speculative ones and returned the rest to the *nspec* seed set. This had the added benefit of providing an opportunity for examining the shortcomings of the substring classifier. The problem with any technique based purely on term matching is the ambiguity inherent in so many natural language expressions. For example, the classifier labels sentences containing the term *potential* as speculative; however, consider the following sentence:

An overview of the local backbone potential is shown in Figure 5.

In this context the nominal use of *potential* does not indicate hedging and the sentence

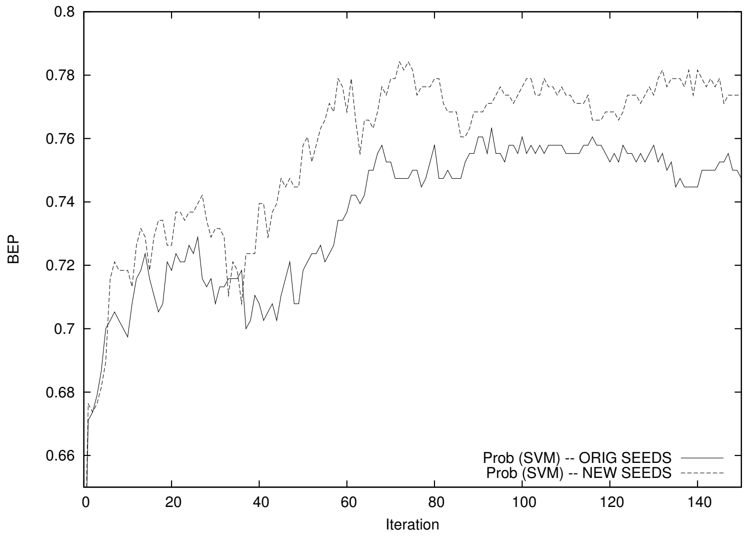


Figure 6.4: Learning curves for new *nspec* seed data

is quite clearly non-speculative. This example also highlights the potential[2] benefit of including part-of-speech information in the representation; an issue we will address later. Part-of-speech information will not always help though; consider a further example:

> *The UAS-brk transgene was amplified from potential mutants by PCR and sequenced.*

It is clear (at least in the authors' opinion) that this sentence is non-speculative and that the adjectival use of *potential* is not a hedge but rather part of the experimental description. Contrast this with:

> *The transient cmg transcription in midgut and Malpighian tubules may suggest a potential function in cell junction formation and in epithelial tissue patterning.*

where *potential* is quite clearly used as a hedge. This illustrates our opinion that simple pattern matching is unlikely to provide a satisfactory solution to the hedge classification task.

We spent around 2-3 hours refining the *nspec* seeds and succeeded in removing 260 spurious instances, yielding a new *nspec* seed set of 7281 *nspec* sentences. Running the probabilistic acquisition model and SVM classifier using the new *nspec* seeds yielded the results shown in Figure 6.4. There is an improvement of around 1-2% BEP.

6.12.3 Exploring *nspec* Acquisition

Examining the sentences chosen by our acquisition model for augmenting the *nspec* training data reveals a noticeable homogeneity of form. Figure 6.5 shows a batch of sentences chosen by the learner for the *nspec* class on the 15th training iteration. It is clear that almost all of these are descriptions of experimental methodology, and as such exhibit certain

[2] no pun intended

T cells were cultured in Dulbecco's modified Eagle medium DMEM Sigma supplemented with...

Abdomens were dissected hr after heat shock from control flies hs-GAL and UAS-IMDhs-GAL flies

FP HA-Rca overexpression using the arm Gal driver line

DfL was generated by mobilizing P-element line l

Ovaries were dissected and labeled with BrdU as described previously

TUNEL staining kits used were the In Situ Cell Death Detection Kit Fluorescein Roche

P1 Diagnostics GmbH and the In Situ Cell Death Detection Kit TMR

All mutagenesis was carried out using the QuickChange mutagenesis kit Stratagene...

RECEIVED MAY REVISED JULY ACCEPTED AUGUST PUBLISHED OCTOBER

Horseradish peroxidase conjugated goat anti-mouse Calbiochem was used as secondary antibody...

After immunostaining samples were treated with DAPI at BCgml for minutes rinsed and mounted

Homozygous CAGSry + hs Cvar- cDNA-EGFP females were crossed to BD m A and Bw mm males...

Egg chambers were then fixed stained and embedded as previously described

Fly expressing a-s nuclein alone aged to da Genotypes wDdc-GAL UAS- -synUAS-HspAL

All peptides were added as solutions in assay medium

Embryos were fixed in methanol for h at room temperature

Molecular Studies- Genomic DNA was isolated from flies using the QIAamp Tissue Kit from...

For latencies each fly was given single pulses

No markers from chromosome five dot chromosome were sequenced

M hep r double-mutant clone arrow bright green and M + clone dark marked with asterisk

Scale bar A BC m in A and B BC m in C and D and BC m in EJ P1

Inserts on chromo somes and were balanced using InLRCyO and InLRTM respectively

The zoo blot was washed for h at BC with C SSC SDS

The antibody was used at for immunohistochemistry and for western blotting

FRT mutant males were crossed to ywP HS-Flp FRT B P arm-lacZ females

Figure 6.5: *nspec* sentences chosen at 15th training iteration (some truncation)

common features, such as past tense predicate constructions. It seems likely that adding increasing numbers of methodology sentences to the *nspec* training data is less than optimal in terms of modelling the *nspec* class, as this would appear to result in a rather unbalanced and misrepresentative training set (there are, after all, many *nspec* sentences in other sections). Thus, we hypothesised that it might be helpful to remove a significant proportion of the methodology sentences from the unlabeled pool in an attempt to force the learner to choose a wider range of *nspec* sentence types. Hence, we removed all of the methodology sentences that could be identified automatically from the source paper markup, 25,244 in total, and re-ran the learning process. In actuality, this approach had little effect on overall accuracy, as shown Figure 6.6 (designated by 'Filtered pool (1)').

Next we tried a more radical approach, extracting from the unlabeled pool only those sentences that were automatically identified as coming from one of the following sections: *Summary*, *Introduction*, *Discussion*, *Results* and *Conclusions*, leaving 108,694 sentences in the pool. These are the sections in which speculations are most likely to be made (Mercer & Marco 2004), and the idea is that the learner chooses *nspec* sentences that are of a similar type to the *spec* ones, thus giving the classifier more chance of discriminating between the difficult instances. Of course, the danger is that the learner is also more likely to acquire spurious *nspec* sentences. Experimental results show that this approach does not improve performance, in fact overall classification accuracy degrades slightly – Figure 6.6, 'Filtered pool (2)'. It is possible that in this scenario a significant proportion of useful *spec* sentences are also removed from the pool, which may contribute to the decrease in performance.

Finally, we also experiment with a scenario in which the *nspec* training data is fixed in its initial seed state and only the *spec* training set is augmented. It is interesting that this has only a marginal negative impact on performance (Figure 6.6 – 'Fixed nspec') which suggests that a relatively small amount is learned about the *nspec* class through the acquisition process, beyond the information already contained in the seed data.

All in all, none of the alternative *nspec* acquisition schemes were able to significantly improve upon the original.

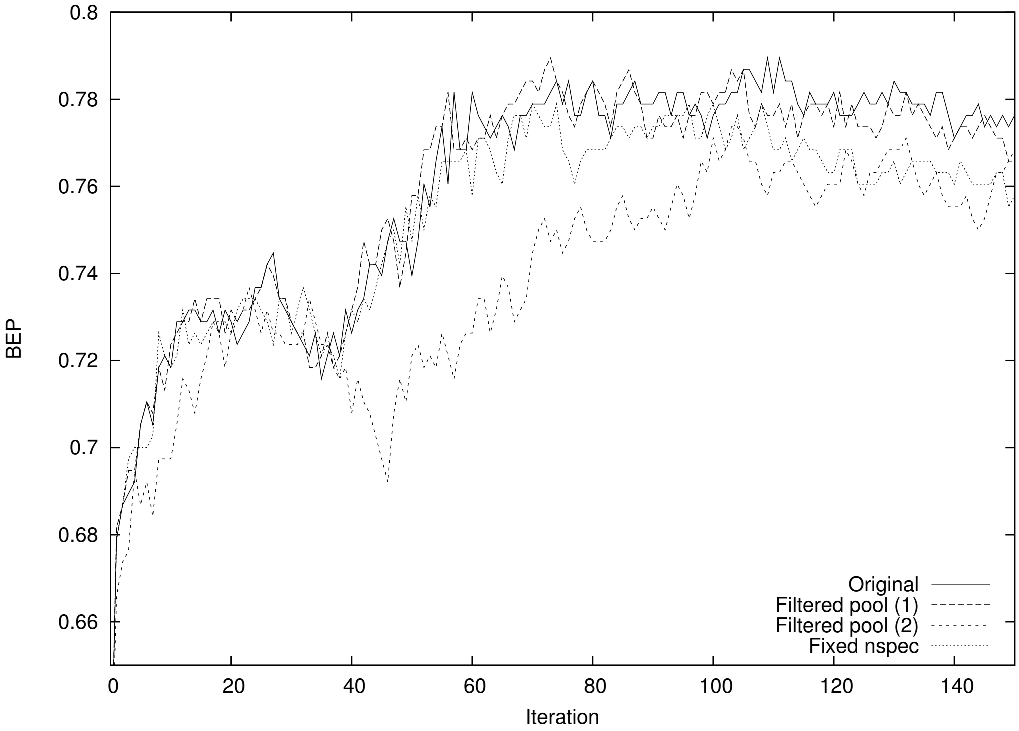


Figure 6.6: Learning curves – exploring *nspec* acquisition

6.12.4 Co-Training

Using the techniques presented thus far, it is not difficult to envisage a co-training extension to the learning model. As discussed earlier, co-training was introduced by Blum & Mitchell (1998) and required a partitioning of the data into two conditionally independent 'views'. It is not clear how such a partition could be generated in our case, bearing in mind the inherent sparsity of relevant features; however Goldman & Zhou (2000) introduced an alternative co-training approach in which two different learning models are used in place of the data partition. When the pool labeling phase is complete, they combine the predictions of the two models to yield the final classification hypothesis. There is a clear analogy for this approach in our distinct acquisition/classification setting, which is to combine the data sets induced by the two acquisition models. Utilising this idea, we propose the following co-training method:

1. Given
   - learning models L_1 (probabilistic) and L_2 (SVM-committee)
   - respective unlabeled pools $\mathcal{U}_1 = \mathcal{U}_2$
   - respective training sets $\mathcal{T}_1^{spec}$, $\mathcal{T}_1^{nspec}$ and $\mathcal{T}_2^{spec}$, $\mathcal{T}_2^{nspec}$
2. Generate seed training data: $\mathcal{S}^{spec}$ and $\mathcal{S}^{nspec}$
3. Initialise: $\mathcal{T}_i^{spec} \leftarrow \mathcal{S}^{spec}$ and $\mathcal{T}_i^{nspec} \leftarrow \mathcal{S}^{nspec}$ $(i = 1, 2)$
4. Iterate:
   - Order $\mathcal{U}_1$ by L_1 trained on $\mathcal{T}_1^{spec}$ and $\mathcal{T}_1^{nspec}$ (*spec* posterior)
   - $\mathcal{T}_2^{spec} \leftarrow$ highest ranked batch
   - $\mathcal{T}_2^{nspec} \leftarrow$ lowest ranked batch

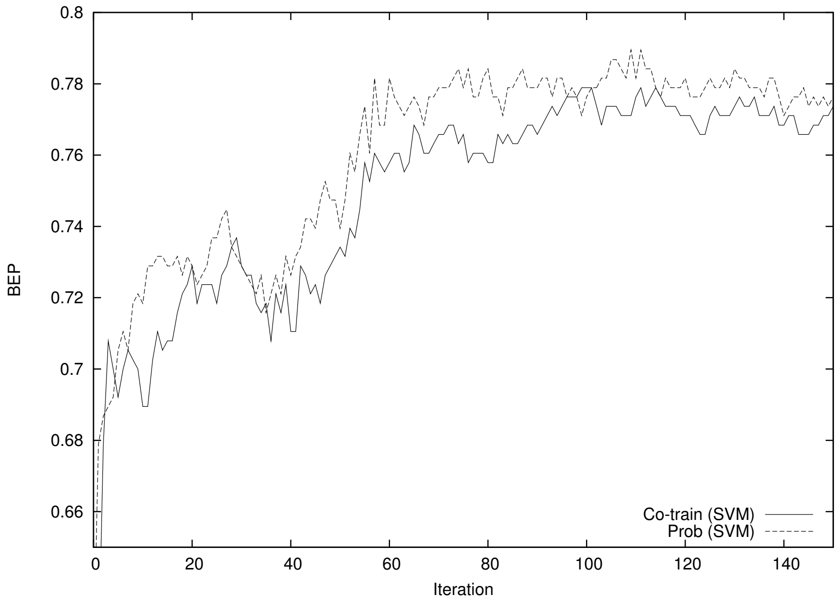


Figure 6.7: Learning curve for co-training model

- Order $\mathcal{U}_2$ by L_2 trained on $\mathcal{T}_2^{spec}$ and $\mathcal{T}_2^{nspec}$ (cumulative confidence)
- $\mathcal{T}_1^{spec} \leftarrow$ highest ranked batch
- $\mathcal{T}_1^{nspec} \leftarrow$ lowest ranked batch

- Combine $\mathcal{T}_1^{spec}$ and $\mathcal{T}_2^{spec} \leftarrow \mathcal{T}_{spec}$
- Combine $\mathcal{T}_1^{nspec}$ and $\mathcal{T}_2^{nspec} \leftarrow \mathcal{T}_{nspec}$

- Train classifier using $\mathcal{T}_{spec}$ and $\mathcal{T}_{nspec}$
- Compute *spec* recall/precision BEP on the test data

Figure 6.7 displays the results of applying our co-training method. There is in fact no improvement over the probabilistic acquisition/SVM classification model, rather overall co-training accuracy is slightly inferior, and disadvantageously the approach is a great deal more expensive than the alternatives proposed in this study, especially those based on the probabilistic acquisition model.

6.12.5 PoS Tagging and Stemming

As suggested above, there is a theoretical motivation for using part-of-speech tags to enrich the sample representation from the point of view of sense disambiguation. We tagged each word using the RASP PoS component, based on a sequential HMM tagger and the CLAWS2 tagset.[3] The results for the augmented representation are given in Figure 6.8. The addition of PoS tags does yield slightly better accuracy in later training iterations than the basic term-based representation, but the improvements are marginal and not statistically significant. In practice, the benefits derived from PoS tags in terms of word sense disambiguation are not as pronounced as theory might suggest. For example, earlier we argued that the term *potential* when used as an adjective is much more likely to represent hedging than when used as a nominal. While this is undoubtedly true, our test

[3] www.comp.lancs.ac.uk/ucrel/claws2tags.html

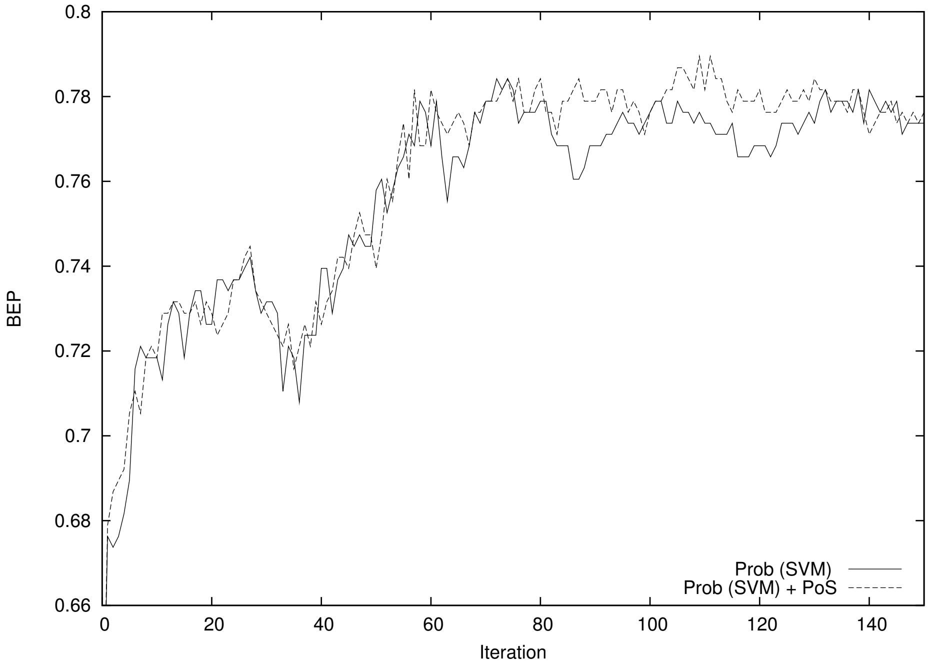


Figure 6.8: Learning curves for PoS tagged representation

data contains no instances of the nominal form of *potential*, and both the *spec* and *nspec* sentences contain the same number of adjectival instances (five). Consider the following:

> *There was considerable excitement in the field when potential mammalian and Drosophila homologs for ced- and egl- were discovered.*

The annotators decided that the use of *potential* in this instance did not represent an authorial hedge because potentiality is by necessity a property of homology.[4] This exemplifies the notion that whether a particular term acts as a hedge cue is quite often a rather subtle function of its sense usage, in which case the distinctions may well not be captured by PoS tagging.

We also experimented with stemming (using the Porter stemmer[5]). The motivation for stemming in hedge classification is that distinct morphological forms of (particularly verbal) hedge cues are often used to convey the same semantics, for instance:

> *Thus these data suggest that dpp signaling interacts with the retinal determination pathway.*

and

> *There is a certain amount of evidence suggesting that dpp signaling interacts with the retinal determination pathway.*

both convey clear speculation through variants of the root verb *suggest*. Verbal forms of nominal hedge cues (and vice-versa) and collapsed in this representation, so for instance

[4] Biological homology refers to structural similarity resulting from shared ancestry, which cannot be established beyond question due to inherent lack of observability; thus is only ever 'potential'.

[5] http://www.tartarus.org/martin/PorterStemmer

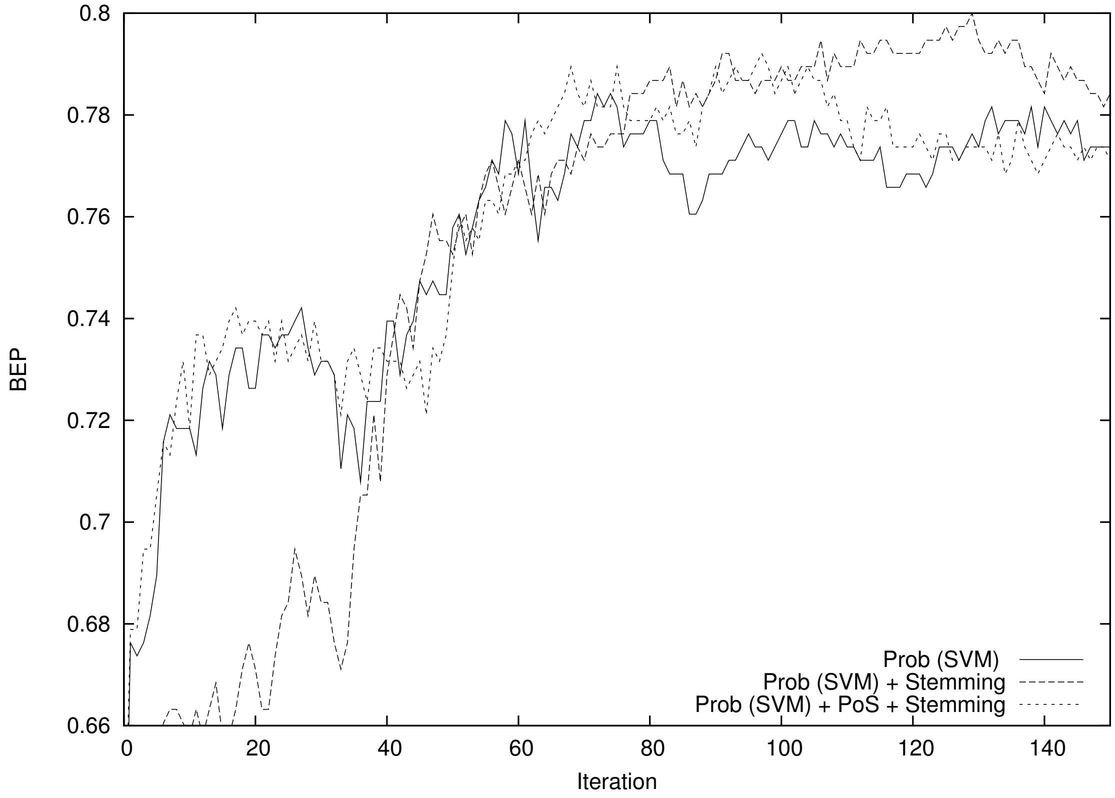


Figure 6.9: Learning curves for stemmed representations

hypothesis and *hypothesize* are both reduced to *hypothesi*. We generate representations that use both stemming on its own (including case normalisation) and in combination with PoS tagging. Figure 6.9 shows that the combined representation follows much the same pattern as the original, which is unsurprising given that PoS tagging has the effect of nullifying the generalisation achieved by stemming. For example, there are separate PoS tags for different verb tenses, which is precisely the sort of information that is discarded by stemming. The interesting case is when stemming is used alone. Over early training iterations, the accuracy of the classifier is significantly lower; however performance continues to improve in latter iterations, yielding a peak result of around 0.8 BEP. Carrying out a binomial sign test comparing the performance of the original and stemmed representations around their relative peaks (80 training iterations for the original representation and 120 for the stemmed variant) showed a weakly significant improvement ($p < 0.2$) for the stemmed representation.

6.12.6 Bigrams

Thus far we have made the assumption that hedge cues are single terms. In reality there are many instances where a hedge cue can be thought of as consisting of more than just one term. For instance, consider the following sentence:

> *In addition several studies indicate that in mammals the Rel proteins could probably be involved in CNS processes such as neuronal development and synaptic plasticity*

Analysis reveals that *'indicate that'* is a fairly reliable hedge cue, whereas *indicate* on its own is not, because of instances such as the following:

> *In the row marked dgqa the stippled exons indicate regions that are not found in the dgqa cDNAs identified by us.*

This suggests that bigram features may be useful, and could potentially enhance the sample representation. Using bigrams results in a well known explosion of the feature space ($O(n) \rightarrow O(n^2)$) and this often prohibits their usefulness due to issues of data sparsity (see arguments in Chapter 3 § 3.5.4). However the hedge classification problem possesses some characteristics that work to its advantage in this regard. Because the number of hedge cues is relatively small, the explosion occurs mostly in the space of *irrelevant* features, and with a reasonably large amount of data we would expect to see the same hedge constructions occuring often enough to yield at least fairly reliable statistics. However, from a semi-supervised learning perspective we must also bear in mind that enriching the features will tend to reduce feature redundancy. Almost all of the research into complex feature generation has concluded that improvements are only gained through *combining* bigrams and single terms (Tan et al. 2002, Moschitti & Basili 2004, Bekkerman & Allan 2005). This has the added advantage that in our case such a scheme is guaranteed to at least retain the redundancy of the original representation, and almost certainly to increase it.

We use the best performing stemmed representation from the previous section, generate all adjacent bigrams and combine them with the single terms. An example of a sentence and its representation is as follows:

> *Several lines of evidence suggest that upregulation of RD gene expression by dpp and ey is likely to account for the synergy that we have observed.*

> sever line of evid suggest that upregul of rd gene express by dpp and ey is like to account for the synergi that we have observ sever_line line_of of_evid evid_suggest suggest_that that_upregul upregul_of of_rd rd_gene gene_express express_by by_dpp dpp_and and_ey ey_is is_like like_to to_account account_for for_the the_synergi synergi_that that_we we_have have_observ

In this representation we include all adjacent bigrams and allow the learning models to select (explicitly or implicitly) the relevant ones. The results are shown in Figure 6.10, and demonstrate that including bigrams yields a clear improvement in accuracy across most of the acquisition curve, with a new peak performance of around 0.82 BEP. According to the binomial sign test, this indicates a statistically significant improvement over both the original representation ($p < 0.01$) and the previous best performing stemmed representation ($p < 0.1$). Table 6.4 shows the 120 highest ranked features according to $P(spec|x_k)$ in the combined single term / bigram representation after 100 learning iterations. There are 13 single terms and 107 bigrams, and it is interesting to note that in some cases neither of the constituent single terms in a given bigram is a likely hedge cue while the combined feature clearly is; for instance *'not_known'* (rank 112), which is a cue for the knowledge paucity hedge.

We also experimented with using dependency features (grammatical relations) derived from RASP (1.3) to augment the single term representation. The motivation for this is to normalise semantically inconsequent surface differences in hedge cue forms. As a simple example consider the following:

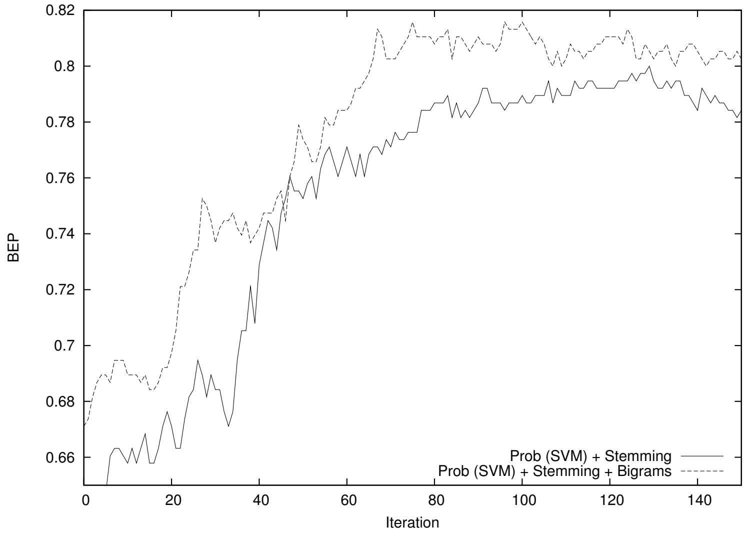


Figure 6.10: Learning curves for stemmed + bigram representations

1	suggest	31	may_not	61	is_unlik	91	unlik_that
2	suggest_that	32	idea_that	62	ask_whether	92	togeth_these
3	might	33	be_due	63	which_may	93	it_might
4	may_be	34	it_may	64	like	94	be_more
5	possibl_that	35	most_like	65	it_appear	95	more_like
6	might_be	36	result_indic	66	whether_thi	96	be_requir
7	appear_to	37	and_may	67	on_possibl	97	unlik
8	result_suggest	38	it_seem	68	we_suggest	98	thei_may
9	propos_that	39	hypothesi_that	69	studi_suggest	99	examin_whether
10	is_like	40	suggest_the	70	not_appear	100	suggest_to
11	thought_to	41	been_suggest	71	appear	101	these_observ
12	suggest_a	42	the_hypothesi	72	suggest_by	102	may_function
13	thi_suggest	43	we_propos	73	might_have	103	suggest_an
14	seem_to	44	test_whether	74	taken_togeth	104	may_act
15	whether_the	45	possibl	75	support_the	105	thu_it
16	whether	46	specul	76	unlik_to	106	that_these
17	data_suggest	47	that_may	77	a_possibl	107	may_contribut
18	like_to	48	observ_suggest	78	been_propos	108	gene_may
19	like_that	49	strongli_suggest	79	evid_suggest	109	which_suggest
20	may_have	50	possibl_is	80	be_a	110	al_suggest
21	may_also	51	rais_the	81	protein_may	111	there_may
22	seem	52	appear_that	82	propos_to	112	not_known
23	may	53	also_be	83	also_suggest	113	is_unclear
24	the_possibl	54	ar_thought	84	play_a	114	and_appear
25	thought	55	and_suggest	85	might_also	115	hypothesi
26	determin_whether	56	be_involv	86	may_play	116	be_respons
27	ar_like	57	thi_may	87	that_might	117	seem_like
28	is_possibl	58	propos	88	find_suggest	118	or_whether
29	is_thought	59	specul_that	89	idea	119	reflect_a
30	the_idea	60	a_role	90	may_reflect	120	to_act

Table 6.4: Single term + bigram features ranked by $P(spec|x_k)$ with $\alpha = 5$.

1. *There is no evidence that C3g affects the enzymatic function of hgg.*
2. *There is no clear evidence that C3g directly affects the enzymatic function of hgg.*

In the first case, the hedge cue is *'no evidence'* while in the second it is *'no clear evidence'*. The important relationship is between the negative determiner *no* and the nominal *evidence*, but the simple adjacency bigram technique presented will not directly capture this, whereas a dependency parser (hopefully) would. Consider the grammatical relations generated by RASP for the two example input sentences:

1. (|ncsubj| |be+s_VBZ| |There_EX| _)
   (|xcomp| _ |be+s_VBZ| |evidence_NN1|)
   (|det| |evidence_NN1| |no_AT|)
   (|ccomp| |that_CST| |evidence_NN1| |affect+s_VVZ|)
   (|ncsubj| |affect+s_VVZ| |C3g_MC| _)
   (|dobj| |affect+s_VVZ| |function_NN1|)
   (|det| |function_NN1| |the_AT|)
   (|ncmod| _ |function_NN1| |enzymatic_JJ|)
   (|iobj| |function_NN1| |of_IO|)
   (|dobj| |of_IO| |hgg_NN1|)

2. (|ncsubj| |be+s_VBZ| |There_EX| _)
   (|xcomp| _ |be+s_VBZ| |evidence_NN1|)
   (|det| |evidence_NN1| |no_AT|)
   (|ncmod| _ |evidence_NN1| |clear:4_JJ|)
   (|ccomp| |that_CST| |evidence_NN1| |affect+s_VVZ|)
   (|ncsubj| |affect+s_VVZ| |C3g_MC| _)
   (|ncmod| _ |affect+s_VVZ| |directly_RR|)
   (|dobj| |affect+s_VVZ| |function_NN1|)
   (|det| |function_NN1| |the_AT|)
   (|ncmod| _ |function_NN1| |enzymatic_JJ|)
   (|iobj| |function_NN1| |of_IO|)
   (|dobj| |of_IO| |hgg_NN1|)

Each line represents a grammatical dependency relationship between two or more terms. In both cases the determiner-nominal relationship between *no* and *evidence* has been identified by the parser (shown in bold). The hope is that other significant non-local dependencies will also be captured, yielding a new set of potential hedge cues that can be exploited by the learning algorithms.

We construct features from the grammatical relations by discarding relationship type and morphological information, and combining the terms in the order they appear. As before, we combine the dependency bigrams with the stemmed single term representation, for example:

> *Several lines of evidence suggest that upregulation of RD gene expression by dpp and ey is likely to account for the synergy that we have observed.*
>
> sever line of evid suggest that upregul of rd gene express by dpp and ey be like to account for the synergi that we have observ suggest_line suggest_upregulation upregul_that upregul_of of_expression express_by by_be be_and be_likely to_likely_account account_for for_synergy synergi_the that_synergy_observe observ_we observ_have and_dpp and_ey express_RD express_gene line_Several line_of of_evidence

We also used pre-filtering to isolate particular GRs thought to be relevant for hedge classification (such as indicators of complementization, e.g. *xcomp* and *ccomp*). Overall, we found that using dependency bigrams in this way resulted in somewhat poorer accuracy than using just the stemmed single terms alone. To understand why this is the case, consider the highest ranking features in Table 6.4 for the adjacent bigram representation. It can be seen that bigrams such as *'suggest that'*, *'possible that'* and *'propose that'* are

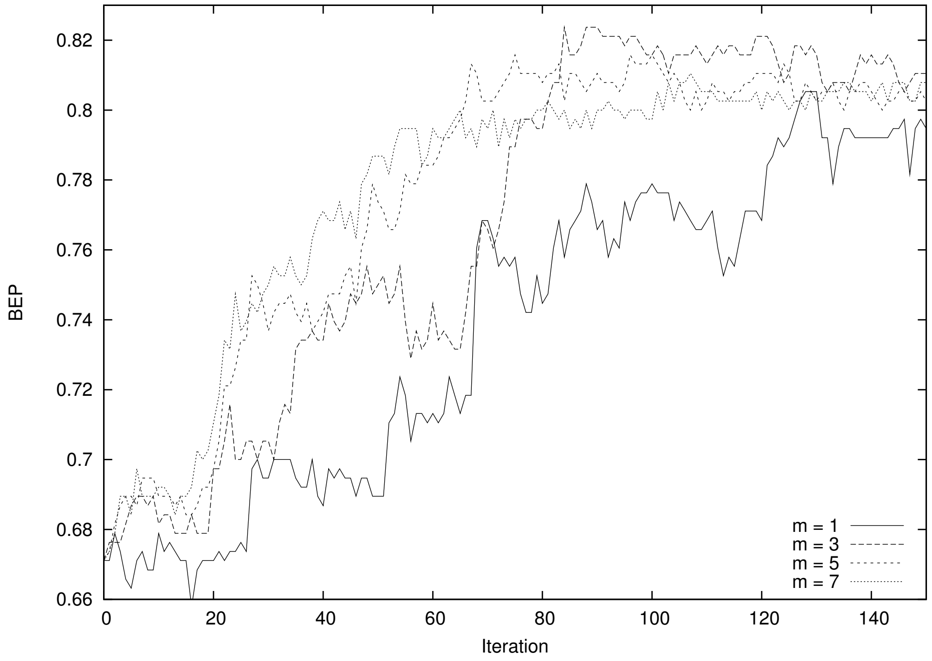


Figure 6.11: Learning curves for stems+bigrams; differing values of m

among the most important hedge cues as they indicate the use of a speculative term with a clausal complement. Now consider the following sentence, shown with the GRs that relate to the hedge term 'suggest':

> *These results suggest that elevation of Dl expression is triggered by the external signal Bnl.*
>
> (|ncsubj| |suggest_VV0| |result+sx_NN2| _)
> (|ccomp| _ |suggest_VV0| |trigger+ed_VVN|)

The important GR here is the second one – 'ccomp' – as it identifies the occurrence of the verb 'suggest' with a clausal complement. However, under our current scheme, the bigram 'suggest_trigger' would be generated from this GR, which only indirectly captures the verb-complement relationship and does not generalise well. A more useful feature would be something like 'suggest_ccomp' which identifies the occurrence of 'suggest' with a clausal complement. To generate features of this type would require the engineering of a more specialized feature set, designed to extract information of particular relevance to the hedge classification task. This is beyond the scope of our current work, but a potential avenue for future research.

6.12.7 Exploring Acquisition Parameters

In all experiments thus far we have used $m = 5$ as the value of the 'feature selection' parameter for the acquisition model. As discussed earlier (§6.10.2), this was chosen on the basis of intuition and a degree of empirical analysis. Here we experiment with different values of m to gauge its impact on performance across the whole of the learning cycle. We use the best performing representation from previous experiments (stems + adjacency

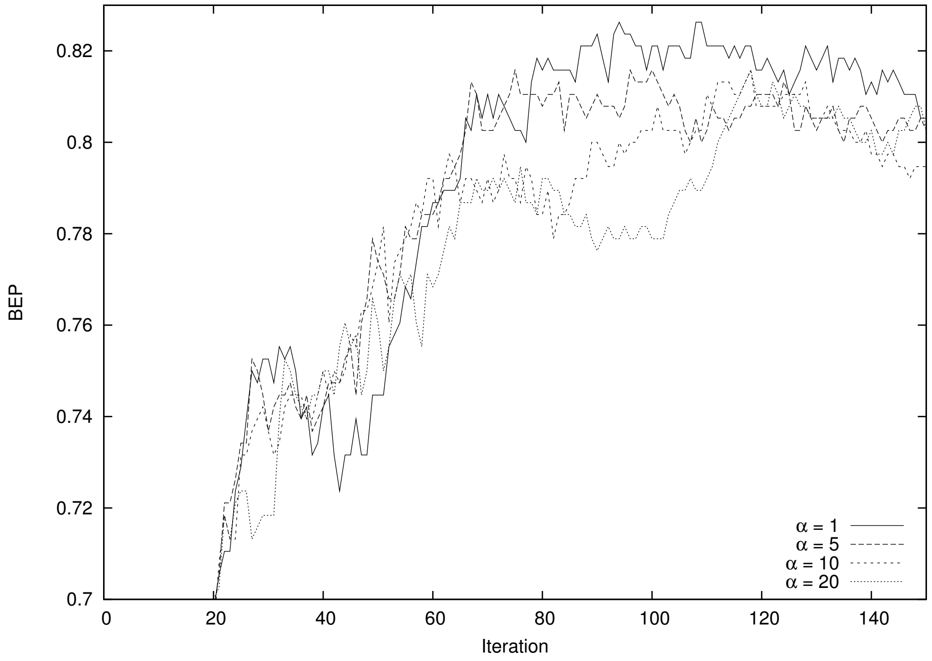


Figure 6.12: Learning curves for stems+bigrams; differing values of α

bigrams) with the probabilistic acquisition model and SVM classifier, and vary m, using the values 1,3,5 and 7. Figure 6.11 shows the results of these experiments; $m = 1$ yields the lowest overall accuracy, representing the scenario in which the learner chooses samples based on the single best looking hedge cue in each sentence. Higher values of m all perform quite similarly, with $m = 3$ yielding marginally better results than $m = 5$, and $m = 7$ slightly worse. The fact that there is no consistent statistically significant difference between the values 3, 5 and 7 after 80-100 training iterations suggests that the acquisition model is fairly robust to the choice of m, with an optimal value of around 3.

We also experimented with different values of the smoothing parameter α: 1, 5 (used in previous experiments), 10 and 20. Figure 6.12 plots the learning curves using the same experimental setup as before (with the default value of $m = 5$ for the feature selection parameter). The results show that $\alpha = 1$ is marginally the best performer, with overall accuracy gradually degrading for higher values. Again, the performance of the acquisition model is quite robust to the choice of α. These results suggest that $\alpha = 1$ may be a better default than $\alpha = 5$, though there is no strong statistical significance between them. Note that our previous empirical analysis (§6.10.2) showed that significantly higher values of α yield observably poorer feature estimates.

6.13 Error Analysis

We examined the errors made by the SVM classifier after 100 iterations of the probabilistic acquisition model using the stem + adjacency bigram sample representation ($m = 5$, $\alpha = 5$). A BEP of 0.816 was obtained at this stage, equating to 310 correctly classified instances out of 380 for the *spec* class and 1087 out of 1157 for the *nspec* class (70 misclassified instances in each class).

A significant proportion (approx. 20%) of the missed *spec* instances were statements of knowledge paucity. These ranged from quite common forms, eg:

> *The role of the roX genes and roX RNAs in this process is still unclear.*

to more unusual variants, eg:

> *This brings us to the largest of all mysteries, namely how the DCC is spread along the X chromosome.*

Such instances are further in construction from the *spec* seed sentences and thus somewhat harder to acquire training data for. A possible way of capturing these instances would be to include specific knowledge paucity seeds.

Some of the missed *spec* instances were due to cases where speculativity is indicated by a particular term, while the general construction of the sentence does not fit the usual *spec* mold. For example:

> *We then tested the putative RNA-binding property of MOF directly using electromobility shift assays.*

This instance looks much like a typical 'materials and methods' sentence, except that the use of *putative* renders it speculative (in the annotators' opinion).

In some cases, genuine hedge cues were not induced with enough certainty, leading to missed *spec* instances, for example:

> *Invertebrates in vivo RAG-mediated transpositions are strongly suppressed, probably to minimize potential harm to genome function.*

The term *probably* is actually a fairly reliable hedge cue, but it only appears at rank 1,268 in the list of features ranked according to $P(spec|x_k)$, estimated from the automatically acquired training data.

Quite a number of missed *spec* instances were just hard to classify, for example:

> *Mutants that pupariated usually showed typical GFP expectoration indicating the presence of a high premetamorphic peak of ecdysteroids.*

It could certainly be argued that this is in fact an instance of observed non-universal behaviour, rather than a hedge. Another example is the following:

> *Some of the intermediate stages of RAG evolution can be inferred from analysis of the sea urchin in which RAG-like proteins were recently observed, and from analysis of the lancelet starlet sea anemone and hydra genomes.*

This instance could be interpreted as a sort of 'meta speculation', stating that speculations about RAG evolution could be made from recent experimental findings. However, it is unclear as to whether this should constitute a hedge in itself.

The majority of false positives (*nspec* instances labeled as *spec*) were due to constructions that are hard to distinguish due to similarity of form, for example:

> *IAPs were first discovered in baculovirus but have since been shown to play a vital role in blocking apoptosis in Drosophila as well as in mammals.*

Variants of the phrase *'play a role'* are quite often used as hedge cues, and hence this instance looks like a hedge, though the annotators decided that in fact it isn't.

Another example of confusion due to similarity of construction is the following:

> *Three Drosophila BIRPs have been shown to be inhibitors of apoptosis Diap Diap and Deterin.*

The infinitive *'to be'* and the verb *be* are in general quite reliable hedge cues (ranked 142 and 217 respectively) whereas in this instance they are not used to signal speculativity. This is also a potential indicator of the disadvantage of combining single terms and bigrams in terms of feature independence violation (though our results show that the benefits outweigh the disadvantages).

In some cases, the classifier actually identified spuriously labeled instances, for example the following were labeled by the annotators as *nspec* when they clearly contain hedges:

> *Caspases can also be activated with the aid of Apaf, which in turn appears to be regulated by cytochrome c and dATP.*

and

> *Further insight into a possible mechanism for IAP function was recently gained when IAPs were observed to have ubiquitin ligase activity.*

We found that around 10% of the false positives were actually due to spurious manual labeling, though for the sake of prior results compatibility we did not carry out any relabeling.

6.14 Conclusions and Future Work

We have shown that semi-supervised ML is applicable to the problem of hedge classification and that a reasonable level of accuracy can be achieved. The work presented here has application in the wider academic community; in fact a key motivation in this study is to incorporate hedge classification into an interactive system for aiding curators in the construction and population of gene databases (Karamanis, Lewin, Seal, Drysdale & Briscoe 2007).

We have presented our initial results on the task in the hope that this will encourage others to investigate this task further. Some potential avenues for future research that we have identified are as follows:

- *Active Learning*: given a classifier trained on acquired data, a profitable subsequent step would be to apply active learning to further augment the training data with instances about which the classifier is uncertain. The combination of semi-supervised and active learning has been explored in various contexts, eg. (McCallum & Nigam 1998, Muslea, Minton & Knoblock 2002), and careful consideration would need to be given to how best to combine the different learning models. It is our intuition that applying semi-supervised and active learning in series may be the best approach for our setting, rather than the more common method of combining them in parallel. The syntactic clustering active learning method presented in §5.5.2 might be a good candidate for this task.

- *Alternative learning models*: it would be interesting to apply a different learning model to the problem, for example *label propagation* as a variant of the semi-supervised paradigm. This would also facilitate the application of existing methods of combining semi-supervised and active learning in the graph based framework, for instance (Zhu, Lafferty & Ghahramani 2003).

- *Representation*: There are various possibilities for enriching the sample representation, perhaps to take account of context, eg. which section of the paper a given sentence was drawn from, and whether its surrounding sentences are speculative. Explicit inclusion of negation might also be beneficial for improving recall of knowledge paucity hedges.

- *Acquisition phase stopping criteria*: an issue we haven't addressed is that of whether the acquisition model can be automatically stopped at an optimal, or close to optimal point. Various methods have been investigated to address this problem, such as 'counter-training' (Yangarber 2003) and committee agreement thresholding (Zhang 2004); more work is needed to establish whether these or related ideas can be applied in our setting.

- *Assertion level hedge classification*: as mentioned earlier, rather than just knowing whether or not a sentence contains a hedge, it would be beneficial to know which assertion a given hedge scopes over. We propose that a sensible method would be to perform further analysis on the (relatively small) subset of sentences identified as belonging to the *spec* class to find the assertion boundaries and the scope of likely hedge cues. This would probably require a degree of syntactic analysis which could be derived from a dependency parser such as RASP.

Chapter 7

Conclusions

7.1 Contributions

Here we look at the research questions listed in §1.2 and consider how effectively they have been answered by the work presented in this report:

Is category separability a reliable correlate for classification accuracy and can it be used as a guide to classifier selection?

This question is addressed in chapter 3. Our experimental results give strong evidence to support the hypothesis that category separability is a good correlate for classification accuracy. We also show that the improvement in accuracy obtained by using more complex classification techniques is related to the separability of the categories. Thus, we contend that category separability analysis is a useful tool both for exploring the nature of a particular classification problem and also for guiding classifier selection.

In applied NLP/classification, is it more important to focus on the sample representation or the machine learning model?

Results from chapter 3 suggest that if the sample representation is chosen carefully, differences in machine learning model performance are minimal, at least in the area of topic classification. In general, we have found that as the amount of training data is increased, the choice of sample representation becomes increasingly important. For complex models it is necessary to choose the representation such that the problem is not made intractable due to a profusion of features; for simpler models it is necessary to choose the representation to highlight key discriminative features and minimise noise. Our conclusion is that this question cannot be answered directly because the type of representation used *depends* on the machine learning model selected and vice versa.

Is there a correlation between sample resolution and the utility of complex features? If so, why?

We have expended a significant amount of effort in this report examining issues of sample representation (3.5, 4.4.2, 5.4.2, 5.6.2, 6.12.5, 6.12.6). Our results support the principle that representation must be tailored to the specific application, and to some extent the choice of classification model. In the areas we have explored, we have found that in general, classification tasks with lower resolution (e.g. sentence or phrase level) are more amenable to deriving benefit from features with greater complexity (term combinations, syntactic structure etc.), while single term 'bag of words' representations are hard to improve upon for high resolution (e.g. document level) tasks. Single terms represent finer grained 'measurements' in samples consisting of many terms (e.g. documents), and thus potentially provide a better level of description than in samples consisting of just a few terms, where expanding the feature space can yield a more thorough description.

Of course, the effectiveness of a particular representation is strongly dependent on the nature of the task in question, as well as on the sample resolution. For instance, bigrams are more effective in hedge classification than topic classification, partly due to differences in sample space resolution, i.e. the relatively high number of category indicative terms in a document renders a 'bag of words' representation quite effective as a description, while a sentence may contain only a few indicative terms, and thus a bag of words representation may be too coarse. However, the disparity is also partly due to the nature of the tasks, i.e. in the case of hedge classification the space of hedge cue terms is quite constrained, and in turn this constrains the space of category indicative bigrams, thus easing the problem of higher order feature sparsity (§6.12.6). Issues of representation must be examined for each task encountered, as they are central to the effectiveness of any classification model.

Are linguistically-motivated features, especially those derived from a 'deep' syntactic analysis of text, useful for classification?

Our attempts to incorporate linguistically motivated components within the classification models met with varying, though on the whole limited, success (3.5.1, 4.3, 5.4.2, 5.6.2, 6.12.5, 6.12.6). In each case we have taken care to explain why the approach did or did not result in improvement, through examination of the nature of the relevant task. Our results cannot be taken as evidence that linguistically-motivated features are irrelevant for classification, but neither have we found an area in which such features are unquestionably beneficial. In this sense we have been unable to provide a clear answer to the above question, either affirmative or contradictory, though we hope to have improved the general level of understanding.

The general consensus of previous research is that linguistically-motivated features are not useful for classification. If our work corroborates this view, why is this the case?

The results of our work support the idea that for some language based classification tasks, linguistically-motivated features are unnecessary, as in the case of topic classification (chapter 3) where independent single terms carry sufficient semantic value to obtain high levels of accuracy given sufficient data. In other cases we have found that simple-minded techniques for generating complex features are effective, as in the case of hedge classification (chapter 6) where using surface-level adjacent bigrams yields significant improvements over single terms, while syntactically-derived bigrams do not. As stated above, we do not believe that our work corroborates the view that linguistically-motivated features are irrelevant for classification in general, rather we have found that preconceptions about the semantic nature of classification tasks do not always hold true (3.5.4, 6.12.6).

Can the task of textual anonymisation be formulated in a fashion that is both amenable to NLP technologies and useful for practical purposes?

Given our presentation and analysis of the textual anonymisation task in chapter 5, we answer this question in the affirmative, but with a measure of caution. Investigating the anonymisation task leads to many complicated issues related to the subjectivity and variability of the nature of the problem. We have shown that, given a certain understanding of the task, NLP techniques can be used to significantly reduce the amount of human input required to anonymise a sizeable dataset; however it remains to be seen whether techniques such as the ones we have presented will be accepted in a real-world setting.

Can the task of sentence-level hedge classification be specified so as to achieve high agreement amongst independent annotators?

In chapter 6 we answer this question in the affirmative by showing that a very high level of agreement ($> 0.95\ \kappa$) can be achieved given a well-specified set of guidelines regarding the identification of hedging in scientific literature (6.5).

Can a high level of accuracy be achieved on the hedge classification task using semi-supervised machine learning?

Again in chapter 6 we demonstrate that greater than 80% precision/recall break-even-point (BEP) can be obtained on the hedge classfication task using semi-supervised machine learning techniques (6.12.6). This represents a high level of accuracy, though there is room for significant improvement, and we hope that this will be achieved through improved training data acquisition, increased understanding of the task and application of better-adapted machine learning techniques.

7.1.1 Specific Aims

The work presented in this report fulfills the specific aims listed earlier (1.2), restated here along with the section in which they are addressed:

- Develop a method for quantifying category separability and examine its impact on classification accuracy. (3.3)
- Explore the use of linguistically motivated features in fine grained topic classification. (3.5.4)
- Develop a new state-of-the-art content-based spam filtering model, taking account of the semi-structured nature of email. (4.4)
- Construct a new anonymised spam filtering corpus with a sizeable proportion of heterogeneous genuine email messages. (4.3)
- Present the problem of anonymisation as a reference level classification task. (5.3)
- Develop annotation schemes and a publicly available corpus of informal text to facilitate anonymisation experiments. (5.4)

- Develop an interactive learning model for anonymisation, motivated by the subjectivity of the task. (5.5.2)
- Specify the problem of hedge classification as a sentence-level classification task. (6.3)
- Develop new annotation guidelines for identifying and labeling hedging in scientific literature. (6.3)
- Assemble a sizeable corpus of biomedical text for hedge classification. (6.4)
- Investigate the properties of the hedge classification task from a semi-supervised machine learning perspective. (6.6)
- Develop and explore a probabilistic model for training data acquisition. (6.7, 6.11, 6.12)

7.2 Directions for Future Research

Finally, we turn to the question of the direction of future research into classification methods for NLP. Which issues are likely to be of particular concern over the coming months and years? Can past and current trends be used to estimate a general direction for the field?

The range of classification techniques applied to NLP tasks is ever increasing, with new variants perpetually introduced as they filter down from the machine learning and statistics communities. A charge sometimes levelled at the current state of research is that there is often a potentially unhealthy focus on the application of new techniques resulting in marginal performance improvements, at the expense of real progression in understanding of the tasks. If this is indeed a genuine concern, a partial remedy may be for the field to favour research that pays close attention to the advantages of newly proposed techniques in terms of real-world applicability and relevance to the task in question, rather than favouring the presentation of fashionable machine learning techniques and statistically 'significant' but practically insignificant intrinsic evaluation metric improvements.

In speculating about the future direction of classification research for NLP, it is worth examining the general nature of NLP problems. It is well known that the distribution of occurrence of words and word senses in natural language can be approximated by a *Zipfian distribution* (Jurafsky & Martin 2000)[1]. In fact the distributions of many lingustic and non-linguistic phenomena are approximately Zipfian (Li 2002). For example, consider the frequency distribution of bigram hedge cues using the *spec* training data from Chapter 6 (6.12.6). Figure 7.1 plots the 100 most frequent bigram hedge cue frequencies against rank, and from the shape of the curves it can be seen that the distribution is approximately Zipfian. An important characteristic of power law distributions (of which the Zipfian is an example), and one which for reasons of space limitation is not fully portrayed by these graphs, is the 'long tail' of rare instances, each of which occurs only a relatively few times in the data.

[1] A Zipfian distribution is one that follows *Zipf's Law* and belongs to the family of *power law* distributions, characterised by the approximate inverse proportionality of instance frequency to rank.

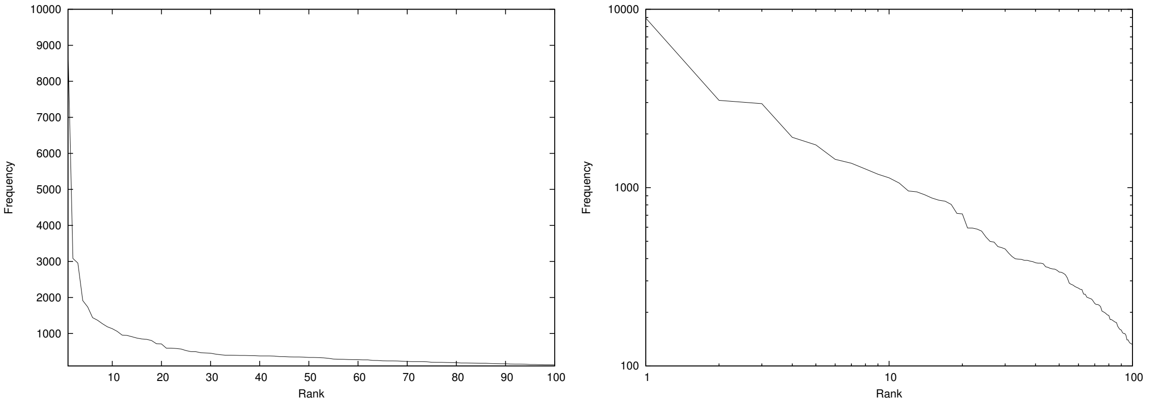


Figure 7.1: Bigram hedge cue frequency distributions (standard and log)

A related phenomenon, though not as easy to visualise, occurs in relation to the complexity of the classification function. In essence, the complexity of the classification function usually grows according to a power law with respect to the intrinsic accuracy of the classifier. For instance, approximately 40% of hedged sentences can be identified using just four single-term hedge cues (*suggest, likely, may* and *might*) without significant loss of precision, but to reach 60% recall at a similar level of precision (approx. 80% BEP) requires a vastly more complex classification function involving the weighted combination of thousands of features (6.12.6). To improve accuracy further, the classifier must attack the instances in the 'long tail' of the distribution, and in the same way that rare language terms tend toward greater complexity, so these rare instances tend to be more complex in structure and thus harder to classify than the samples forming the bulk of the distribution. This means that not only does the quantity of training data required to improve classification accuracy grow exponentially with respect to standard performance metrics (Banko & Brill 2001), the complexity of the classification function must also increase at a concomitant rate, requiring a potentially herculian increase in complexity to accurately classify the most difficult instances.

The upshot of these considerations is that while state-of-the-art classification performance is relatively good, it is our opinion that radically different models and representations may be required to reach a level of performance across the range of tractable NLP tasks that could be considered equivalent to that of a human expert. Of course this depends on the nature and complexity of the task; some problems are generally considered to be practically solved – English part-of-speech tagging for instance – but current solutions to the majority of NLP classification tasks still fall significantly short of the ideal, and raising the accuracy by just a few percent is deceptively challenging for the reasons sketched above.

At the same time we must continually ask whether our evaluation strategies actually reflect the real world utility of NLP systems, and explore methods for carrying out realistic extrinsic evaluation. To sum up, in our opinion the current situation leaves plenty of headroom for future classification research, as ever increasing volumes of data facilitate exploration into more interesting representations and models. At the same time, consistent reevaluation of explorative and evaluative methodology will lead to a deeper understanding of both problem and solution.

Bibliography

Abney, S. (2002), Bootstrapping, *in* 'Proceedings of 40th Annual Meeting of the Association for Computational Linguistics', pp. 360–367.

Amati, G., Carpineto, C. & Romano, G. (2004), Query difficulty, robustness, and selective application of query expansion, *in* 'ECIR', pp. 127–137.

Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C. & Stamatopoulos, P. (2000), 'Learning to filter spam email: A comparison of a naive bayesian and a memorybased approach', *Workshop on Machine Learning and Textual Information Access.*

Apte, C., Damerau, F. & Weiss, S. M. (1994), 'Automated learning of decision rules for text categorization', *Information Systems* **12**(3), 233–251.

Artstein, R. & Poesio, M. (2005), Kappa^3 = alpha (or beta), Technical report, University of Essex Department of Computer Science.

Banko, M. & Brill, E. (2001), Scaling to very very large corpora for natural language disambiguation, *in* 'Meeting of the Association for Computational Linguistics', pp. 26–33.

Bekkerman, R. & Allan, J. (2005), Using bigrams in text categorization, Technical Report 408, CIIR.

Bennett, P. N. (2003), Using asymmetric distributions to improve text classifier probability estimates, *in* 'SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval', ACM Press, New York, NY, USA, pp. 111–118.

Bennett, S. W., Aone, C. & Lovell, C. (1997), Learning to tag multilingual texts through observation, *in* C. Cardie & R. Weischedel, eds, 'Proceedings of the Second Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Somerset, New Jersey, pp. 109–116.

Berger, A. L., Pietra, V. J. D. & Pietra, S. A. D. (1996), 'A maximum entropy approach to natural language processing', *Comput. Linguist.* **22**(1), 39–71.

Bishop, C. M. (2006), *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Blum, A. & Mitchell, T. (1998), Combining labeled and unlabeled data with co-training, *in* 'COLT' 98: Proceedings of the eleventh annual conference on Computational learning theory', ACM Press, New York, NY, USA, pp. 92–100.

Borko, H. & Bernick, M. (1963), 'Automatic document classification.', *J. ACM* **10**(2), 151–162.

Borko, H. & Bernick, M. (1964), 'Automatic document classification. part ii: additional experiments', *Journal of the Association for Computing Machinery* **11**(2), 138–151.

Borthwick, A. E. (1999), A maximum entropy approach to named entity recognition, PhD thesis. Adviser-Ralph Grishman, New York University, New York, NY, USA.

Bouma, L. & de Rijke, M. (2006), Specificity helps text classification, *in* 'ECIR', pp. 539–542.

Bratko, A. & Filipič, B. (2004), Exploiting structural information in semi-structured document classification, *in* 'Proc. 13th International Electrotechnical and Computer Science Conference, ERK'2004'.

Briscoe, T., Carroll, J. & Watson, R. (2006), The second release of the rasp system, *in* 'Proceedings of the COLING/ACL on Interactive presentation sessions', Association for Computational Linguistics, Morristown, NJ, USA, pp. 77–80.

Bruckner, M. & Dilger, W. (2005), A soft bayes perceptron, *in* 'Proceedings of the IEEE International Joint Conference on Neural Networks, IJCNN '05'.

Cappé, O., Moulines, E. & Ryden, T. (2005), *Inference in Hidden Markov Models (Springer Series in Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Carreras, X. & Marquez, L. (2001), 'Boosting trees for anti-spam email filtering', *Proceedings of RANLP2001* pp. 58–64.

Chang, C.-C. & Lin, C.-J. (2001), *LIBSVM: a library for support vector machines.* Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Chao, G. & Dyer, M. G. (2002), Maximum entropy models for word sense disambiguation, *in* 'Proceedings of the 19th international conference on Computational linguistics', Association for Computational Linguistics, Morristown, NJ, USA, pp. 1–7.

Chen, J., Ji, D., Tan, C. L. & Niu, Z. (2006), Relation extraction using label propagation based semi-supervised learning, *in* 'Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics', Association for Computational Linguistics, pp. 129–136.

Chen, S. & Goodman, J. (1996), 'An empirical study of smoothing techniques for language modeling', *Proceedings of the 34th Annual Meeting of the ACL.*

Clarke, R. (1997), 'Privacy and dataveillance, and organisational strategy', *Proceedings of the Region 8 EDPAC'96 Information Systems Audit and Control Association Conference.*

Cohen, W. W. (1995), Text categorization and relational learning, *in* A. Prieditis & S. J. Russell, eds, 'Proceedings of ICML-95, 12th International Conference on Machine Learning', Morgan Kaufmann Publishers, San Francisco, US, Lake Tahoe, US, pp. 124–132.

Cohn, D. A., Ghahramani, Z. & Jordan, M. I. (1995), Active learning with statistical models, *in* G. Tesauro, D. Touretzky & T. Leen, eds, 'Advances in Neural Information Processing Systems', Vol. 7, The MIT Press, pp. 705–712.

Collins, M. & Singer, Y. (1999), Unsupervised models for named entity classification, *in* 'Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999'.

Cormack, G. & Lynam, T. (2005), Spam corpus creation for TREC, *in* 'Proceedings of Second Conference on Email and Anti-Spam CEAS 2005'.

Cormack, R. (1971), 'A review of classification', *Journal of the Royal Statistical Society. Series A (General)* **134**(3), 321–367.

Corti, L., Day, A. & Backhouse, G. (2000), 'Confidentiality and informed consent: Issues for consideration in the preservation of and provision of access to qualitative data archives', *Forum for Qualitative Social Research.*

Cronen-Townsend, S. & Croft, W. B. (2002), Quantifying query ambiguity, *in* 'Proceedings of the second international conference on Human Language Technology Research', Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 104–109.

Cronen-Townsend, S., Zhou, Y. & Croft, W. B. (2002), Predicting query performance, *in* 'SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval', ACM, New York, NY, USA, pp. 299–306.

Damashek, M. (1995), 'Gauging similarity with n-grams: Language-independent categorization of text', *Science* **267**(5199), 843–848.

Debole, F. & Sebastiani, F. (2004), An analysis of the relative hardness of Reuters-21578 subsets, *in* 'Proceedings of LREC-04, 4th International Conference on Language Resources and Evaluation', Lisbon, PT, pp. 971–974.

Denoyer, L. & Gallinari, P. (2004), 'Bayesian network model for semi-structured document classification.', *Inf. Process. Manage.* **40**(5), 807–827.

Diederich, J., Kindermann, J., Leopold, E. & Paass, G. (2003), 'Authorship attribution with support vector machines', *Applied Intelligence* **19**(1-2), 109–123.

Drucker, H., Wu, D. & Vapnik, V. (1999), 'Support vector machines for spam categorization', *IEEE Trans. On Neural Networks* **10(5)**, 1048–1054.

Duda, R. O. & Hart, P. E. (1973), *Pattern Classification and Scene Analysis*, John Willey & Sons, New York.

Dumais, S., Platt, J., Heckerman, D. & Sahami, M. (1998), Inductive learning algorithms and representations for text categorization, *in* 'CIKM '98: Proceedings of the seventh international conference on Information and knowledge management', ACM Press, New York, NY, USA, pp. 148–155.

ESDS (2004), 'The economic and social data service: Identifiers and anonymisation: dealing with confidentiality', `http://www.esds.ac.uk/aandp/create/identguideline.asp`.

Field, B. (1975), 'Towards automatic indexing: automatic assignment of controlled-language indexing and classification from free indexing', *Journal of Documentation* **31**(4), 246–265.

Finn, A., Kushmerick, N. & Smyth, B. (2002), Genre classification and domain transfer for information filtering, *in* 'Proceedings of the 24th BCS-IRSG European Colloquium on IR Research', Springer-Verlag, London, UK, pp. 353–362.

Forman, G. & Cohen, I. (2004), Learning from little: Comparison of classifiers given little training., *in* 'PKDD', pp. 161–172.

Freund, Y. & Schapire, R. E. (1998), Large margin classification using the perceptron algorithm, *in* 'Computational Learning Theory', pp. 209–217.

Fuhr, N. (1989), 'Models for retrieval with probabilistic indexing', *Inf. Process. Manage.* **25**(1), 55–72.

Fuhr, N. & Buckley, C. (1991), 'A probabilistic learning approach for document indexing', *ACM Trans. Inf. Syst.* **9**(3), 223–248.

Fuhr, N. & Pfeifer, U. (1994), 'Probabilistic information retrieval as combination of abstraction, inductive learning and probabilistic assumptions', *ACM Transactions on Information Systems* **12**(1), 92–115.

Genkin, A., Lewis, D. D. & Madigan, D. (2005), 'Large-scale bayesian logistic regression for text categorization'.

Ghahramani, Z. (2005), 'Non-parametric bayesian methods', UAI '05 Tutorial.

Goldman, S. & Zhou, Y. (2000), Enhancing supervised learning with unlabeled data, *in* 'Proc. 17th International Conf. on Machine Learning', Morgan Kaufmann, San Francisco, CA, pp. 327–334.

Hamill, K. A. & Zamora, A. (1980), 'The use of titles for automatic document classification', *Journal of the American Society for Information Science* **33**(6), 396–402.

He, B. & Ounis, I. (2006), 'Query performance prediction', *Inf. Syst.* **31**(7), 585–594.

Hripcsak, G. & Rothschild, A. (2004), 'Agreement, the f-measure, and reliability in information retrieval', *J Am Med Inform Assoc.* **12**(3), 296–298.

Hull, D. A., Pedersen, J. O. & Schütze, H. (1996), Method combination for document filtering, *in* H.-P. Frei, D. Harman, P. Schäuble & R. Wilkinson, eds, 'Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval', ACM Press, New York, US, Zürich, CH, pp. 279–288.

Hyland, K. (1994), 'Hedging in academic writing and eap textbooks', *English for Specific Purposes* **13**, 239–256.

Jaynes, E. T. (1957), 'Information theory and statistical mechanics', *Physical Review* **106**(4), 620+.

Jelinek, F. & Mercer, R. (1980), 'Interpolated estimation of markov source parameters from sparse data', *Proceedings of the Workshop on Pattern Recognition in Practice.*

Joachims, T. (1997), A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization, *in* D. H. Fisher, ed., 'Proceedings of ICML-97, 14th International Conference on Machine Learning', Morgan Kaufmann Publishers, San Francisco, US, Nashville, US, pp. 143–151.

Joachims, T. (1998), Text categorization with support vector machines: learning with many relevant features, *in* 'Proceedings of the European Conference on Machine Learning (ECML 98)', number 1398, Springer Verlag, Heidelberg, DE, Chemnitz, DE, pp. 137–142.

Joachims, T. (1999), Making large-scale support vector machine learning practical, *in* A. S. B. Schölkopf, C. Burges, ed., 'Advances in Kernel Methods: Support Vector Machines', MIT Press, Cambridge, MA.

Joachims, T. (2006), Training linear svms in linear time, *in* 'KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM Press, New York, NY, USA, pp. 217–226.

Jones, K. S. (1991), 'Notes and references on early automatic classification work.', *SIGIR Forum* **25**(1), 10–17.

Jurafsky, D. & Martin, J. H. (2000), *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice Hall PTR, Upper Saddle River, NJ, USA.

Kambhatla, N. (2004), Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations, *in* 'Proceedings of the ACL 2004 on Interactive poster and demonstration sessions', Association for Computational Linguistics, Morristown, NJ, USA, p. 22.

Karamanis, N., Lewin, I., Seal, R., Drysdale, R. & Briscoe, T. (2007), Integrating natural language processing with FlyBase curation, *in* 'Proceedings of PSB 2007', pp. 245–256.

Katz, E. (1987), 'Estimation of probabilities from sparse data for the language model component of a speech recognizer', *IEEE Transactions on Acoustics Speech and Signal Processing* **35(3)**, 400–401.

Kessler, B., Nunberg, G. & Schütze, H. (1997), Automatic detection of text genre, *in* P. R. Cohen & W. Wahlster, eds, 'Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics', Association for Computational Linguistics, Somerset, New Jersey, pp. 32–38.

Kilgarriff, A. & Rose, T. (1998), Measures for corpus similarity and homogeneity, *in* 'Proceedings of the 3rd conference on Empirical Methods in Natural Language Processing', Granada, Spain, pp. 46–52.

Kim, S., Rim, H., Yook, D. & Lim, H. (2002), Effective methods for improving naive bayes text classifiers, *in* 'Proc. 7th Pacific Rim International Conference on Artificial Intelligence', Vol. 2417, Springer, Heidelberg, pp. 414–42.

Klimt, B. & Yang, Y. (2004), The enron corpus: A new dataset for email classification research., *in* 'Proceedings of the European Conference on Machine Learning (ECML 04)', pp. 217–226.

Krauth, W. & Mezard, M. (1987), 'Learning algorithms with optimal stability in neural networks', *Journal of Physics A: Math. Gen.*

Lafferty, J. D., McCallum, A. & Pereira, F. C. N. (2001), Conditional random fields: Probabilistic models for segmenting and labeling sequence data, *in* 'ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning', Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 282–289.

Lam, W. & Ho, C. Y. (1998), Using a generalized instance set for automatic text categorization, *in* 'SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval', ACM Press, New York, NY, USA, pp. 81–89.

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

Lewis, D. D. (1992), An evaluation of phrasal and clustered representations on a text categorization task, *in* 'SIGIR '92', ACM Press, New York, NY, USA, pp. 37–50.

Lewis, D. D., Yang, Y., Rose, T. G. & Li, F. (2004), 'Rcv1: A new benchmark collection for text categorization research', *J. Mach. Learn. Res.* **5**, 361–397.

Li, W. (2002), 'Zipf's law everywhere', *Glottometrics* **5**, 14–21.

Li, Y., Bontcheva, K. & Cunningham, H. (2005), Using uneven margins SVM and perceptron for information extraction, *in* 'Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)', Association for Computational Linguistics, Ann Arbor, Michigan, pp. 72–79.

Light, M., Qiu, X. & Srinivasan, P. (2004), The language of bioscience: Facts, speculations, and statements in between, *in* 'Proceedings of BioLink 2004 Workshop on

Linking Biological Literature, Ontologies and Databases: Tools for Users, Boston, May 2004'.

Lovis, C. & Baud, R. (1999), 'Electronic patient record: dealing with numbers or with words?', *World Multiconference on Systemics, Cybernetics and Informatics Vol 8* pp. 181–197.

Luhn, H. P. (1957), 'A statistical approach to mechanized encoding and searching of literary information', *IBM Journal of Research and Development* **1**(4), 309–317.

Malouf, R. (2002), A comparison of algorithms for maximum entropy parameter estimation, *in* 'COLING-02: proceeding of the 6th conference on Natural language learning', Association for Computational Linguistics, Morristown, NJ, USA, pp. 1–7.

Maron, M. E. (1961), 'Automatic indexing: An experimental inquiry', *J. ACM* **8**(3), 404–417.

Martínez, D., Agirre, E. & Màrquez, L. (2002), Syntactic features for high precision word sense disambiguation, *in* 'Proceedings of the 19th international conference on Computational linguistics', Association for Computational Linguistics, Morristown, NJ, USA, pp. 1–7.

Masand, B., Linoff, G. & Waltz, D. (1992), Classifying news stories using memory based reasoning, *in* 'SIGIR '92', ACM Press, New York, NY, USA, pp. 59–65.

McCallum, A. & Nigam, K. (1998), 'A comparison of event models for naive bayes text classification', *AAAI-98 Workshop on Learning for Text Categorization.*

McCallum, A. K. (2002), Mallet: A machine learning for language toolkit, http://mallet.cs.umass.edu.

McClosky, D., Charniak, E. & Johnson, M. (2006), Effective self-training for parsing, *in* 'Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics', Association for Computational Linguistics, Morristown, NJ, USA, pp. 152–159.

Medlock, B. (2006*a*), An adaptive, semi-structured language model approach to spam filtering on a new corpus, *in* 'Proceedings of the Third Conference on Email and Anti-Spam (CEAS 2006)'.

Medlock, B. (2006*b*), An introduction to nlp-based textual anonymisation, *in* 'Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)'.

Medlock, B. & Briscoe, T. (2007), Weakly supervised learning for hedge classification in scientific literature, *in* 'Proceedings of the 45rd Annual Meeting of the Association for Computational Linguistics (ACL'07)', Association for Computational Linguistics, Prague.

Mercer, R. E. & Marco, C. D. (2004), A design methodology for a biomedical literature indexing tool using the rhetoric of science, *in* ‘LT-NAACL 2004 Workshop: Biolink 2004, Linking Biological Literature, Ontologies and Databases’.

Mladenic, D. & Grobelnik, M. (1999), Feature selection for unbalanced class distribution and naive bayes, *in* ‘ICML ’99: Proceedings of the Sixteenth International Conference on Machine Learning’, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 258–267.

Moschitti, A. & Basili, R. (2004), Complex linguistic features for text classification: A comprehensive study., *in* ‘Proceedings of the European Conference on Information Retrieval (ECIR)’, pp. 181–196.

Muslea, I., Minton, S. & Knoblock, C. A. (2002), Active + semi-supervised learning = robust multi-view learning, *in* ‘ICML ’02: Proceedings of the Nineteenth International Conference on Machine Learning’, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 435–442.

Ng, V. & Cardie, C. (2003), Weakly supervised natural language learning without redundant views, *in* ‘NAACL ’03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology’, Association for Computational Linguistics, Morristown, NJ, USA, pp. 94–101.

Nigam, K., McCallum, A. K., Thrun, S. & Mitchell, T. M. (2000), ‘Text classification from labeled and unlabeled documents using EM’, *Machine Learning* **39**(2/3), 103–134.

Pang, B. & Lee, L. (2004), A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, *in* ‘Proceedings of the ACL’, pp. 271–278.

Pang, B., Lee, L. & Vaithyanathan, S. (2002), Thumbs up? Sentiment classification using machine learning techniques, *in* ‘Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)’, pp. 79–86.

Pedersen, T. (2000), A simple approach to building ensembles of naive bayesian classifiers for word sense disambiguation, *in* ‘Proceedings of the first conference on North American chapter of the Association for Computational Linguistics’, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 63–69.

Peng, F. & Schuurmans, D. (2003), Combining naive bayes and n-gram language models for text classification., *in* ‘Proceedings of the European Conference on Information Retrieval (ECIR)’, pp. 335–350.

Preiss, J. (2003), Using grammatical relations to compare parsers, *in* ‘EACL ’03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics’, Association for Computational Linguistics, Morristown, NJ, USA, pp. 291–298.

Quinlan, J. R. (1993), *C 4.5: Programs for machine learning*, The Morgan Kaufmann Series in Machine Learning, San Mateo, CA: Morgan Kaufmann, 1993.

Rabiner, L. R. (1989), 'A tutorial on hidden markov models and selected applications in speech recognition', *Proceedings of the IEEE* **77**(2), 257–286.

Raina, R., Shen, Y., Ng, A. & McCallum, A. (2004), 'Classfication with hybrid generative/discriminative models', *NIPS 16, 2004*.

Raskutti, B., Ferrá, H. L. & Kowalczyk, A. (2001), Second order features for maximising text classification performance., *in* 'Proceedings of the European Conference on Machine Learning (ECML 2001)', pp. 419–430.

Rasmussen, C. E. & Williams, C. K. I. (2005), *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*, The MIT Press.

Rehm, G. (2002), 'Towards automatic web genre identification – a corpus-based approach in the domain of academia by example of the academic's personal homepage'.

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 (ICML-2003)', AAAI Press, pp. 616–623.

Riloff, E. (1996), Using learned extraction patterns for text classification, *in* S. Wermter, E. Riloff & G. Scheler, eds, 'Connectionist, statistical, and symbolic approaches to learning for natural language processing', Springer Verlag, Heidelberg, DE, pp. 275–289. Published in the "Lecture Notes in Computer Science" series, number 1040.

Rios, G. & Zha, H. (2004), Exploring support vector machines and random forests for spam detection., *in* 'Proceedings of the First Conference on Email and Anti-Spam (CEAS 2004)'.

Robertson, S. E. & Harding, P. (1984), 'Probabilistic automatic indexing by learning from human indexers', *Journal of Documentation* **40**(4), 264–270.

Rock, F. (2001), Policy and practice in the anonymisation of linguistic data, *in* 'International Journal of Corpus Linguistics', Vol. 6, John Benjamins Publishing Company.

Roddick, J. & Fule, P. (2003), A system for detecting privacy and ethical sensitivity in data mining results, Technical Report SIE-03-001, School of Informatics and Engineering, Flinders University.

Rosenblatt, F. (1958), 'The perceptron: A probabilistic model for information storage and organization in the brain', *Psychological Review* **65**, 3862̆013408.

Sahami, M., Dumais, S., Heckerman, D. & Horvitz, E. (1998), 'A bayesian approach to filtering junk e-mail', *Learning for Text Categorization - Papers from the AAAI Workshop* pp. 55–62.

Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C. & Stamatopoulos, P. (2000), 'Stacking classifiers for anti-spam filtering of e-mail.', *Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing (EMNLP 2001)* pp. 44–50.

Schapire, R. E. & Singer, Y. (2000), 'BoosTexter: A boosting-based system for text categorization', *Machine Learning* **39**(2/3), 135–168.

Scholkopf, B. & Smola, A. J. (2001), *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, MA, USA.

Sebastiani, F. (2002), 'Machine learning in automated text categorization', *ACM Computing Surveys* **34**(1), 1–47.

Shen, D., Zhang, J., Su, J., Zhou, G. & Tan, C.-L. (2004), Multi-criteria-based active learning for named entity recognition, *in* 'ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics', Association for Computational Linguistics, Morristown, NJ, USA, p. 589.

Stamatatos, E., Fakotakis, N. & Kokkinakis, G. (1999), Automatic authorship attribution, *in* 'Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics', Association for Computational Linguistics, Morristown, NJ, USA, pp. 158–164.

Sutton, C. & McCallum, A. (2006), An introduction to conditional random fields for relational learning, *in* L. Getoor & B. Taskar, eds, 'Introduction to Statistical Relational Learning', MIT Press. To appear.

Tan, C.-M., Wang, Y.-F. & Lee, C.-D. (2002), 'The use of bigrams to enhance text categorization', *Inf. Process. Manage.* **38**(4), 529–546.

Tang, M., Luo, X. & Roukos, S. (2001), Active learning for statistical natural language parsing, *in* 'ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics', Association for Computational Linguistics, Morristown, NJ, USA, pp. 120–127.

Vapnik, V. & Chervonenkis, A. J. (1974), 'Ordered risk minimization (i and ii)', *Automation and Remote Control* **34**, 1226–1235 and 1403–1412.

Vapnik, V. N. (1995), *The nature of statistical learning theory*, Springer-Verlag New York, Inc., New York, NY, USA.

Wahlstrom, K. & Roddick, J. F. (2001), On the impact of knowledge discovery and data mining, *in* J. Weckert, ed., 'Selected papers from the 2nd Australian Institute of Computer Ethics Conference (AICE2000)', ACS, Canberra, pp. 22–27.

Weiss, S. M., Apte, C., Damerau, F. J., Johnson, D. E., Oles, F. J., Goetz, T. & Hampp, T. (1999), 'Maximizing text-mining performance', *IEEE Intelligent Systems* **14**(4), 63–69.

Wellner, B. (2005), Weakly supervised learning methods for improving the quality of gene name normalization data, *in* 'Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics', Association for Computational Linguistics, Detroit, pp. 1–8.

Whitelaw, C., Garg, N. & Argamon, S. (2005), Using appraisal groups for sentiment analysis, *in* 'CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management', ACM Press, New York, NY, USA, pp. 625–631.

Witten, I. H. & Frank, E. (2002), 'Data mining: practical machine learning tools and techniques with java implementations', *SIGMOD Rec.* **31**(1), 76–77.

Wolters, M. & Kirsten, M. (1999), Exploring the use of linguistic features in domain and genre classification, *in* 'Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics', Association for Computational Linguistics, Morristown, NJ, USA, pp. 142–149.

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

Yang, Y. & Pedersen, J. O. (1997), A comparative study on feature selection in text categorization, *in* D. H. Fisher, ed., 'Proceedings of ICML-97, 14th International Conference on Machine Learning', Morgan Kaufmann Publishers, San Francisco, US, Nashville, US, pp. 412–420.

Yang, Y., Zhang, J. & Kisiel, B. (2003), A scalability analysis of classifiers in text categorization, *in* 'SIGIR '03', ACM Press, New York, NY, USA, pp. 96–103.

Yangarber, R. (2003), Counter-training in discovery of semantic patterns, *in* E. Hinrichs & D. Roth, eds, 'Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics', pp. 343–350.

Yarowsky, D. (1995), Unsupervised word sense disambiguation rivaling supervised methods, *in* 'Proceedings of the 33rd annual meeting on Association for Computational Linguistics', Association for Computational Linguistics, Morristown, NJ, USA, pp. 189–196.

Yi, J. & Sundaresan, N. (2000), A classifier for semi-structured documents, *in* 'KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining', ACM Press, pp. 340–344.

Yom-Tov, E., Fine, S., Carmel, D. & Darlow, A. (2005), Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval, *in* 'SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval', ACM, New York, NY, USA, pp. 512–519.

Zhang, J., Jin, R., Yang, Y. & Hauptmann, A. G. (2003), Modified logistic regression: An approximation to svm and its applications in large-scale text categorization., *in* 'Proceedings of the Twentieth International Conference on Machine Learning (ICML)', pp. 888–895.

Zhang, T. & Oles, F. J. (2001), 'Text categorization based on regularized linear classification methods', *Information Retrieval* **4**(1), 5–31.

Zhang, Z. (2004), Weakly-supervised relation classification for information extraction, *in* 'CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management', ACM Press, New York, NY, USA, pp. 581–588.

Zhao, S. & Grishman, R. (2005), Extracting relations with integrated information using kernel methods, *in* 'ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics', Association for Computational Linguistics, Morristown, NJ, USA, pp. 419–426.

Zhou, G. & Su, J. (2001), Named entity recognition using an HMM-based chunk tagger, *in* 'ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics', Association for Computational Linguistics, Morristown, NJ, USA, pp. 473–480.

Zhu, X., Lafferty, J. & Ghahramani, Z. (2003), Combining active learning and semi-supervised learning using gaussian fields and harmonic functions, *in* 'Proc. of the ICML 2003 Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining'.