Abstract

Current Information Retrieval (IR) techniques succeed in identifying relevant documents. However there may be a large number of relevant documents and it is difficult to isolate those documents that most closely match a particular user's information needs. Genre analysis provides a complementary technique that can be used to improve results achieved using standard IR techniques. A genre class is a group of documents that are written in a similar style. Genre classification can be used to identify documents that are written in a style most likely to satisfy a user's information need.

We consider the use of Machine Learning techniques applied to the task of automatic genre classification. We investigate two sample genre classification tasks: whether a news article is subjective or objective; and whether a review is positive or negative. We investigate the use of three different feature-sets for building genre classifiers.

We argue that traditional methods of evaluating text classifiers are insufficient for genre classifiers and emphasize domain transfer for the generated classifiers. Domain transfer indicates the ability of a genre classifier to classify documents that are about topics other than those it was trained on.

Our experiments show that genre classification using Machine Learning techniques is challenging, but feasible. For both sample genre classification tasks, we build classifiers that perform well within a single topic domain. We find that it is difficult to build genre classifiers that transfer well to other domains.

Acknowledgements

This project would not have been possible without the help of a great many people. I am grateful to you all.

Thanks to my supervisors, Nick and Barry for guiding me through the last two years. Thanks to everybody in the Department of Computer Science, especially those who shared an office with me for the past two years.

Thanks to Claire, my confidante. To my family John, Catherine and Ann-Marie.

Most of all I would like to thank my parents, Michael and Lucia, whose unwavering support and encouragement made this possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Information Overload

There is a vast amount of information on the WWW, however the potential of the Web as an instant information source is being hampered. The WWW has resulted in a huge increase in the amount of information available to people and there are now large amounts of information available about any particular subject. However, with this information explosion there comes a significant problem: the information overload problem.

It is becoming increasingly difficult to locate and isolate relevant information. The number of documents relevant to a particular query can be overwhelming. Current techniques succeed in retrieving potentially relevant information, but it is more difficult to isolate highly relevant information. There are likely to be thousands of documents with content relevant to any particular query, and different users issuing the same query will have differing information needs. Current WWW search services take a “one size fits all” approach, which takes little account of the user's individual needs and preferences. Each user's information need may be satisfied by a different subset of the set of documents returned by a search. We believe that topic alone is insufficient for ranking document relevance.

1.2 Genre Classification

The genre of a document reflects the style of language used in the document. Genre is a difficult term to define. When we use the term genre, we are referring to 'text genre'. The text genre of a particular document is a category to which the document belongs based on some aspect of the style of text used in the document. A genre class is a group of documents written in a similar style. The set of genre classes for a particular document collection should have some useful discriminatory function for the users of the document collection. Once an appropriate set of genre classes have been identified, we seek methods for automatically identifying them.

Genre is orthogonal to topic. Documents that are about the same topic can be from different genres (Figure 1.1). Similarly, documents from the same genre can be about different topics. Thus we must separate the identification of the topic and genre of a document and try to build genre classifiers that are topic independent. When we evaluate our classifiers, we attempt to measure how well the classifier performs across multiple topic domains.

The term "genre" occurs frequently in popular culture. Music is divided into genres based on differences in style, e.g. blues, rock or jazz. Sample genres from popular fiction include science fiction, mystery and drama. Genres

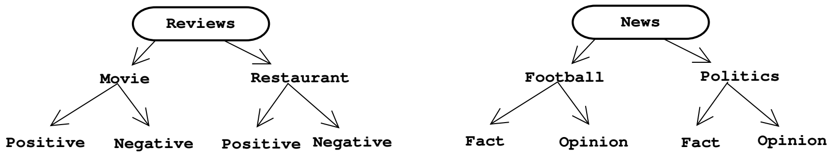

Figure 1.1: Genre is orthogonal to topic

are often vague concepts with no clear boundaries and need not be disjoint. For a given subject area there are is no fixed set of genre categories. Identifying a genre taxonomy is a subjective process and people may disagree about what constitutes a genre, or the criteria for membership of a particular genre.

The American heritage dictionary of the English language defines genre as "A category of artistic composition, as in music or literature, marked by a distinctive style, form or content". Webster's revised unabridged dictionary defines a genre as "class; form; style esp. in literature". Wordnet defines genre as "1: a kind of literary or artistic work 2: a style of expressing yourself in writing 3: a class of artistic endeavour having a characteristic form or technique".

From the different definitions we see that there is no definitive agreement on what is meant by genre. However, the common thread among these definitions is that genre relates to style. The genre of a document reflects a certain style rather than being related to the content. In general this is what we mean when we refer to the genre of a document: the genre describes something about what kind of document it is rather than what the document is about.

To summarize, we view a genre as a class of documents that arises naturally from study of the language style and text used in the document collection. Genre is an abstraction based on a natural grouping of documents written in a similar style and is orthogonal to topic.

The readability of a document could be considered a genre classification. Readability measures [8] can be used to give and indication of the density of text and the level of educational background needed to read it. A tabloid newspaper generally has higher readability than a broad-sheet. Different users searching for information will require different levels of readability.

The level of technical detail in a document can be considered another genre class. An engineer searching for information on a topic would probably require a different document than a journalist searching for information on the same topic. Users generally are not looking for all information on a certain topic but rather certain kinds of information about that topic. By identifying the genre of a document, we can further reduce the set of documents that have relevant content to a set of documents that have both relevant content and are from genres that are appropriate to a users current information need.

We focused on two sample genres which we use for our automatic genre classification experiments. The next two sections describe these two genre classes.

1.2.1 Sample Genre: Opinion or fact

The first genre class we investigate is whether a document presents the opinion of its author or reports facts. This is a common distinction in newspaper articles and other media. Many news articles report some significant event objectively. Other articles offer the author's opinion. These often take the form of columns or editorials. This genre class is particularly useful for filtering news articles because a user may have a preference for articles of each of these genre classes at different times.

Consider the example of financial news. Financial news sites publish many articles each day. Articles of genre class fact may be reporting the latest stock prices and various events that are likely to influence the stock price of a particular company. Articles of genre class opinion may give the opinions of various financial analysts as to the implications of the events of the day for future stock prices. Different users at different times may be better served by articles from one genre or the other. It would be a useful service for the user to be able to filter or retrieve documents from each of these genre classes.

Table 1.1 show a selection of document extracts from three subject domains. A human reader can recognize a subtle difference in style between extracts from articles of genre class opinion and articles of genre class fact. We investigate techniques for automating this classification.

Our aim in constructing the classifier is to maximize accuracy both within a single domain and across domains. To this end we use datasets from three subject domains: football, politics and finance[1]. We are interested in how well a classifier trained on documents from one domain performs in another. We identify the different subject domains in order to determine how well the classifier performs across multiple subject domains. Because genre is orthog-

[1]Our document corpus is available to the research community at http://www.smi.ucd.ie/hyppia

	Fact	Opinion
Football	Liverpool have revealed they have agreed a fee with Leeds United for striker Robbie Fowler - just hours after caretaker boss Phil Thompson had said that contract talks with the player were imminent.	The departure of Robbie Fowler from Liverpool saddens me but does not surprise me. What did come as a shock, though, was that the club should agree terms with Leeds, one of their chief rivals for the Championship.
Politics	Al Gore picked up votes Thursday in Broward County as election officials spent Thanksgiving weekend reviewing questionable presidential ballots.	Democrats are desperate and afraid. The reality that their nominee for President has a compulsive tendency to make things up to make himself look good is sinking in.
Finance	In a move that sent Enron shares higher after days of double-digit declines, Dynegy confirmed Tuesday that it is in talks to renegotiate its $9 billion deal to buy its rival.	The collapse of Enron is hard to believe, and even harder to understand. But in retrospect, there are some valuable lessons in the whole mess.

Table 1.1: Examples of objective and subjective articles from three subject domains.

onal to topic we wish to build classifiers that perform well in subject domains other than the one used to build the classifier. For example, a classifier built to recognize whether a document is subjective or objective by training it on documents about football should ideally be able to recognize subjective documents that are about topics other than football such as finance or politics (figure 1.2).

1.2.2 Sample Genre: Positive or Negative

Our second sample genre class is classifying reviews as being either positive or negative. Table 1.2 shows examples of positive and negative reviews from two subject domains. The subject domains are movie reviews and restaurant reviews. Again we wish to build a classifier that can automatically distinguish between positive and negative reviews across multiple subject domains. A classifier built by training it on reviews from the movie domain should ideally be able to classify restaurant reviews also.

The ability to automatically recognize the tone of a review could have application in the collaborative recommendation systems. For example, if a particular movie critic who generally has similar tastes to a user gives a film a positive review, then that film could be recommended to the user. Films could be recommended on the basis of how they are reviewed by critics that are known to have similar tastes to a particular user.

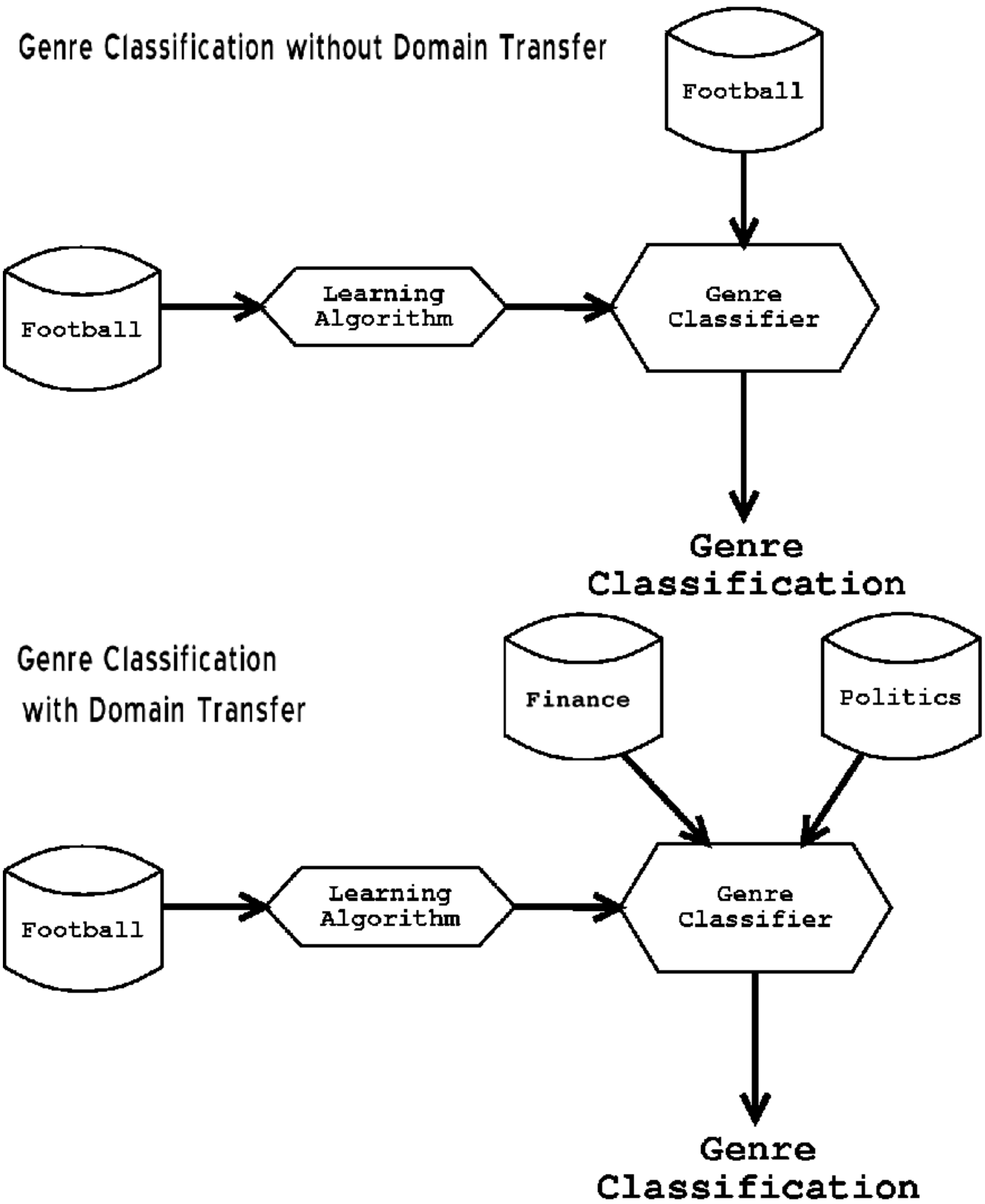

Figure 1.2: Genre Classification for Domain Transfer

	Positive	Negative
Movie	Almost Famous: Cameron Crow's first film since "Jerry Maguire" is so engaging, entertaining and authentic that it's destined to become a rock-era classic. Set in 1973, this slightly fictionalized, semi-autobiographical, coming-of-age story revolves around a baby-faced 15 year old prodigy whose intelligence and enthusiasm land him an assignment from "Rolling Stone" magazine to interview Stillwater, an up-and-coming band.	Vanilla Sky: Presumably Cameron Crowe and Tom Cruise have some admiration for "Abre Los Ojos" the 1998 Spanish thriller from Alejandro Amenabar; why else would they have chosen to do an English-language remake? "Vanilla Sky", however shows that respect for ones source material isn't enough. It's a misbegotten venture that transforms a flawed but intriguing original into an elephantine, pretentious mess.
Restaurant	Though the New American menu at this neighbourhood treasure near Capitol Hill is ever changing, it's always beautifully conceived and prepared and based on mostly organic ingredients; the bistro dishes, paired with a fabulous, descriptive wine list, are served in an offbeat atmosphere.	Hidden in the back of a shopping mall near Emory, this Chinese eatery is so isolated that diners sometimes feel as if they're having a private meal out; the decor isn't much to look at and the foods nothing special but it's decent.

Table 1.2: Examples of positive and negative reviews from two subject domains

1.3 Content-Based Analysis

Current techniques use the topic of a document for retrieval and recommendation. While topic alone can identify relevant documents, we believe that additional techniques are necessary to find the most relevant documents. The traditional methods of information retrieval using keyword analysis have more recently been augmented with link analysis [23], latent semantic indexing [12] and collaborative filtering [14] to improve retrieval. However genre analysis has not been widely explored. Google uses link analysis to re-order search results and identify high quality information sources. This significantly improves the quality of results returned for a particular query.

We identify genre analysis as a technique for improving search quality. Genre analysis can be used to filter the results of a search and to identify documents most suited to the particular users information need. Genre analysis can facilitate personalization by identifying documents that are most appropriate and interesting to a particular user.

Genre can be used to classify documents that are written in a style relevant to a particular user. In small homogeneous collections of documents, genre may be quite similar across the document collection. However in a large heterogeneous document collection such as the WWW, identifying the genre of a document can significantly contribute to retrieving the most relevant documents for a particular user. Genre analysis is therefore a valuable tool in the construction of personalized retrieval and filtering systems.

1.4 Motivation

There is a vast amount of information available to the casual user today mainly due to the proliferation of the world wide web (WWW). However, in addition to the many benefits this brings, there are also drawbacks. The main drawback is the huge amount of information a user faces and the difficulty in finding the most relevant information. This information overload problem means that while users can usually find relevant information, there is a difficulty in identifying the most relevant information for a particular query. It is easy to achieve high recall, but high precision in addition to high recall is more difficult to achieve. The huge volume of information available means that new techniques are needed to help reduce the volume of material considered relevant and identify the information that best satisfies a users information need.

Current approaches use content analysis to identify the most relevant documents to a given user query [47]. Documents are represented as a set of keywords, and the presence of the query terms in a document would indicate that the document is relevant to that query. The importance of the terms occurring in a document may be analyzed according to some weighting scheme such as TF/IDF [41]. This approach analyses the relevance of a document to a query by attempting to discern the topic of the document. The keyword representation of a document is a shallow representation of the topic of the document. However topic alone is not the only factor which indicates whether a document is relevant or not.

Recently other techniques have been used to augment this process. Collaborative filtering [38, 14] can recommend documents on the basis that other users found the document relevant and those users have similar interests. A document can be deemed relevant to a particular user not on the basis of the content of that document, but rather because it has been found relevant to another user that has similar interests to the target user.

Google uses link analysis [23, 6] to rank documents that have been returned using standard keyword techniques. Documents that are linked to by many other documents are judged to be documents of high value. Initially relevant documents are identified using keyword matching. These results are then ranked using the PageRank [5] algorithm. Intuitively this leads to documents that are linked to by many other documents being ranked highly in the search results. This approach complements the content analysis approach and gives better search results.

Latent Semantic Indexing (LSI) [12] identifies terms that are related to each other by performing a statistical analysis of the entire document space. This allows documents to be retrieved because they contain words that tend to often occur in relation to the terms in the query. LSI produces a lower dimensional vector space whose dimensions are obtained as combinations of the original dimensions by inferring dependence among the original terms.

We have identified automatic genre analysis as an additional tool that can complement existing techniques and improve the results returned to a user. Recall that by genre we mean the type of text in the document. A genre class is a class of documents that are of a similar type. This classification is based not on the topic of the document, but rather on the kind of text used. We believe that genre analysis can be used to filter out documents that are of kinds likely to be uninteresting to the user and identify those documents that are most relevant.

We believe that automatic genre classification has applications in information retrieval, information filtering, personalization, document recommendation and document classification. The ability to filter documents according to text genre would be a valuable service in any text retrieval system. For example, consider a query about "chaos theory". Different users will require documents which assume different levels of expertise, depending on the users technical background. It would be useful to be able to rank documents according to the level of technical detail with which they present their subject. Current information retrieval systems would be greatly enhanced by the ability to filter documents according to their genre class. A high school student may require documents that are introductory or tutorial in style, while a college professor may require scholarly research documents.

As another example, consider news filtering. This is generally done according to the topic of the news article. This service would be improved by the ability to filter the news articles according to different genre classes. For example, consider a financial analyst who tracks daily news about companies in which she is interested. It would be useful to be able to further class these documents as being subjective or objective. One class of documents would present the latest news about the various companies of interest, while the other class would contain the opinions of various columnists and analysts about these companies. Depending on circumstances, the user may require documents of one class or the other.

Personalized news filtering and recommendation systems would be enhanced by the ability to filter documents according to genre class. A set of genre classes could be identified which represent the type of documents that would best serve that user. For example, the ability to filter documents according to the level of technical information presented and the readability of the document would enable a system to personalize documents retrieved according to the user's educational background. With a suitable set of genre classes, a system with a dual category structure, that allowed users to browse documents according to both topic and genre would be useful.

Personalization can take place along several dimensions. Content can be used to recommend documents to users based on their previous interests. Collaborative filtering recommends documents based on the fact that other similar users found them interesting.

Genre analysis can facilitate improved personalization by recommending documents that are written in a style that the user finds interesting. We consider genre to be complimentary to topic as a method of recommendation. The two used in conjunction with each other can improve the quality of a user's recommendations.

The ability to identify whether a document is describing something in a positive or negative way could be used to improve a recommender system. Documents could be recommended on the basis that they were given a positive review by a reviewer with similar interests to the target user.

Another application of review classification is filtering of newswire articles for financial analysis. Financial analysts must quickly digest large amounts of information when making investment decisions. A delay of a few seconds in identifying important information can result in significant gains or losses. The ability to automatically identify whether news about a company is positive or negative would be a valuable service in such a situation.

1.5 Approach

Our approach is to use Machine Learning techniques for automatic genre analysis. We attempt to identify the features which will lead to classifiers that perform well and can easily be built automatically. We investigate the performance of our classifiers across multiple subject domains. We use two sample genre tasks to test the utility of three sets of features for the purpose of automatic genre classification. We compared several Machine Learning algorithms and chose C4.5 as our main learning algorithm.

We investigate three approaches to feature selection, namely bag of words (BOW), part-of-speech (POS) tagging and hand-crafted text statistics (HC). We evaluate these three approaches on two sample genre identification tasks. The first is that of subjectivity classification; i.e. whether a document is subjective or objective. This is a common distinction in news articles. Objective articles report the occurrence of some newsworthy event in an objective manner. Subjective articles often take the form of editorials or columns and give the authors opinion on some event.

The second genre classification task we consider is classifying reviews according to whether they are positive or negative. This has uses in recommender systems.

We emphasize the ability to transfer to new subject domains when building our classifiers and we evaluate different feature-sets for performance across multiple subject domains.

1.6 Evaluation

With a sufficient, appropriate genre taxonomy, genre classification can significantly improve document retrieval and recommendation. Documents can be identified that are from genres that are most appropriate to the particular user. Users can also filter out documents that are from genres they consider inappropriate on uninteresting.

We investigate issues involved in building a genre classifier. We aim to build genre classifiers that are general and re-usable. Using our two sample genre classification tasks, we apply machine learning techniques to the task of building general reusable genre classifiers.

We investigate different feature-sets such as part-of -speech tagging (POS) and bag of words (BOW) and evaluate how well the feature-sets discriminate between genre classes.

For many genre classes, the genre of the document is orthogonal to the topic of the document. Documents of the same genre class can have different topics. Traditional applications of document classification aim to segregate documents into classes based on the subject matter of the documents. Because of the orthogonality of genre and topic, traditional methods of evaluating the quality of a classifier may not be sufficient. We need to identify how well our classifiers perform on documents from subject domains other than those they were trained on. We want to identify how much of our classifiers discriminatory power is topic specific. Because of the effort involved in building a genre classifier, a classifier that is topic specific is less useful because it will not work on documents from subject domains other than those it was trained on. When evaluating our classifiers, it is important to evaluate the classifier's ability to transfer to new unseen subject domains.

When evaluating our classifiers, we introduce a new additional measure. We seek to evaluate the domain transfer of the classifier. This indicates the classifiers performance when it is trained and tested on different subject domains. We use this measure to indicate the classifiers generality and it's ability to classify documents from new subject domains.

For a genre classifier to be useful it should be accurate, transferable, easily built. A classifier that is not accurate is of little use. Any useful genre classifier must at least perform well within the subject domain it was trained on. However in addition to having a classifier that performs well within the subject domain in which it was trained, we also want a genre classifier to be transferable. It should perform reasonably well on documents from subject domains other than the subject domain on which it was trained. In addition the genre classifier should be easy to automatically build. The more human effort that is involved in building a classifier the less practically useful it is. To this end we investigate two feature-sets that can be easily extracted automatically from a document collection and one feature-set that requires human effort in it's construction. We would prefer feature-sets from that are automatically extractable to those that are not, if both performed similarly for accuracy and domain transfer.

1.7 Contributions

Our contributions will be:

- To investigate the feasibility of *genre classification using machine learning*. We wish to investigate whether machine learning can successfully be applied to the task of genre classification. We use two sample genre classification tasks to investigate the issues arising from the task of genre classification.

- To investigate how well *different feature-sets* perform on the task of genre classification. Using two sample genre classification tasks, we perform experiments using three different feature-sets and investigate which features satisfy the criteria for building good genre classifiers.

- To investigate the issues involved in building genre classifiers with *good domain transfer*. The task of genre classification requires additional methods of evaluation. We introduce the idea of domain transfer as an indication of the performance of a genre classifier across multiple subject domains. We evaluate each of the feature-sets for their ability to produce classifiers with good domain transfer.

1.8 Thesis Overview

The remainder of this thesis is organized as follows.

Chapter 2 describes background and related work. We give an overview of background material and previous work in the application of Machine

Learning techniques to Text Classification. We survey the related work in genre classification and relate it to our own.

Chapter 3 describes the document encoding and the features used for learning. We investigate three feature-sets: bag-of-words, part-of speech and hand crafted heuristics. This chapter describes the way we represent these documents using the different feature-sets.

Chapter 4 constitutes the main body of the text and describes the experiments performed along with results and analysis. We describe our experimental setup and detail several experiments that evaluate the performance of the different feature-sets for genre classification.

Finally, chapter 5 presents our conclusions and future work.

Chapter 2

Related Work

2.1 Overview

In this chapter we describe the work related to our research and background from related fields.

We give an overview of the learning process and describe the learning algorithms that we use in our experiments. We describe previous work in Text Classification. We believe that the genre classification methods we describe here have applications in information retrieval, filtering and personalization. We describe some previous work in these fields that use methods other than genre classification.

Our main contribution is in the area of genre classification and we survey related work from this field and the related field of stylistics. For each piece of work surveyed, we compare and relate it to our own work. We discuss the different interpretations of what is meant by the term genre and the different approaches taken to automatically identifying genre. We position our work in relation to other work in the area.

2.2 Machine Learning

In this section we present an overview of the Machine Learning approach to building text classifiers. We describe the Machine Learning algorithms that we used for our experiments. We describe the process for evaluating Machine Learning algorithms and argue that these methods are not wholly sufficient for genre classification tasks. Later in chapter 4 we will introduce a new method of evaluation to measure domain transfer which when combined with the usual methods of evaluation provides a better estimate of the utility of a genre classifier.

2.2.1 The Learning Process

The Machine Learning approach to document classification takes a set of pre-classified examples and uses these to induce a model which can be used to classify future instances. The classifier model is automatically induced by examination of the training examples. The human effort in this process is in assembling the labeled examples and choosing a representation for the training examples. A human must initially decide what features will be used to describe the training examples, and represent the training documents with respect to these features. We describe the features that we use as input to our learning algorithm in chapter 3.

When using Machine Learning algorithms, we first identify the concept to be learned. This concept is what we want the classifier to be able to classify;

in our case this is the particular genre class we are attempting to classify. The output of the learning algorithm is a concept description that should ideally be both intelligible and operational. The concept description should be intelligible in the sense that it can be understood, discussed, disputed and interrogated; operational so that we can practically apply it to future examples.

The type of learning we are interested in is classification learning. In this learning scheme, the learner takes a set of labeled pre-classified examples. The learner is then expected to induce ways of classifying unseen examples based on the pre-classified examples given. This form of learning is supervised in that the training examples are provided and labeled by a human overseer.

The training data is a set of instances. Each instance is a single example of the concept to be learned. Instances are characterized by a set of attributes where each attribute measures a certain aspect of the concept being described. Attributes can be nominal or continuous. continuous attributes represent some numerical value that can be measured. Nominal attributes are categorical. They assign the attribute to membership of a particular category.

Representing the classification problem as a set of instances is a restrictive way of formulating the learning problem. Each instance is characterized by values of a set of predetermined attributes. The selection of these attributes can affect the quality of the classifier produced by the learning algorithm. As part of our experiments, we are interested in identifying attributes which perform well on the genre classification task and can be easily extracted automatically.

We use this Machine Learning approach to build our genre classifiers. We use a set of documents that have been labeled according to text genre as examples to build our classifiers. We identify three different sets of features and investigate the utility of each of these for genre classification.

2.2.2 OneR

OneR is a simple learning algorithm for generating one level decision trees. It is useful as a baseline comparison for C4.5. It can also give a rough estimate of the difficulty of the classification task at hand.

OneR generates a set of rules that classify an example instance based on a single attribute. Each attribute generates a set of rules, one for each value of the attribute. The error rate is evaluated for each attributes rule-set and the rule-set with the lowest error rate is chosen. Figure 2.1 shows the OneR algorithm.

The basic OneR algorithm is easily extended to incorporate missing and numeric values. Missing values are treated as another attribute value. Numeric values are discretized into nominal categories.

Despite it's simplicity, OneR performs well on many classification tasks. Holte [16] showed that OneR does surprisingly well in comparison to many state of the art algorithms on several commonly used datasets. It as on average just a few percent less accurate, but generated substantially simpler models and smaller trees. Over-fitting is likely to occur when attributes have a large number of attribute values.

We use OneR to investigate the difficulty of the classification task and to

```
For each attribute
    For each value of that attribute, make a rule as follows:
        construct a rule that assigns the most frequent class of
        this attribute-value to this attribute-value
    calculate the error-rate of the rules.
Choose the rules with the smallest error rate.
```

Figure 2.1: OneR: An algorithm to induce one level decision trees

gain insight into what attributes are good at discriminating the genre class in question.

2.2.3 C4.5

C4.5 [36] is an algorithm for inducing decision trees based on a set of labeled examples. It was developed by Ross Quinlan as a set of extensions and improvements to the ID3 decision tree induction algorithm. Improvements in C4.5 include:

- Support for missing values.
- Support for numeric attributes.
- Pruning of induced decision trees.
- Rule derivation.

ID3 is an algorithm for top-down induction of decision trees. In the resulting decision tree, each node corresponds to a non-categorical attribute. Each arc from the node corresponds to a possible value of that attribute. Each leaf describes the expected value of the categorical attribute for an instance described by the path from the root node to that leaf.

```
DecisionTreeInduce(attributes,instances):
    Select an attribute to place at the root node
    Make a branch for each possible value
    if all instances have the same classification: break
    For each branch:
        DecisionTreeInduce(attributes not used reaching this branch,
                           instances that actually reach this branch)
```

Figure 2.2: An algorithm for decision tree induction

Each node should be associated with the non-categorical attribute that is most informative among those not yet considered on the path from root to node. The notion of entropy is used to measure how informative an attribute is. In particular a measure of information gain is used to choose the most informative attribute.

Figure 2.2 shows a simple decision tree induction algorithm similar to that used by ID3. It uses a measure of information gain to choose which attribute to split on. The information gain of each attribute is calculates and used to choose the attribute with the most information gain.

We use C4.5 as our main learning algorithm for experiments. We chose C4.5 because it generally performs well and the model produced can be easily examined and interrogated.

2.2.4 Other Learning Algorithms

Another simple method is to use all attributes to calculate the probability of an instance belonging to a category based on the observed training data. This technique is called Naive Bayes [26] and is based on Bayes' rule. It naively assumes independence between attributes. Despite this assumption, naive Bayes works well in practice, particularly when used on datasets that have fairly independent attributes. Naive Bayes can rival or outperform more sophisticated algorithms on many datasets and has been widely used for Text Classification.

Another method for learning classifiers is instance based learning [10]. In this case, the training examples are stored verbatim. A distance function is used to determine which member of the training set is closest to the instance being classified. The class of the nearest neighbor is then used as the predicted class for an instance. Most instance based learners use Euclidean distance for their distance function. The distance between an instance with attribute values $a_1^{(1)}, ...a_k^{(1)}$ and $a_1^{(2)}, ...a_k^{(2)}$ is given by

$$\sqrt{(a_1^{(1)} - a_1^{(2)})^2 + ... + (a_k^{(1)} - a_k^{(2)})^2}$$

This kind of classifier can be corrupted by noisy training examples. One solution to this problem is to use the k-nearest neighbors for classification. For example with k=5, the 5 nearest neighbors of an instance are located and a majority vote is used to determine the classification.

We used these learning algorithms for some initial experiments before choosing C4.5 as our learning algorithm (as explained in section 4.4).

2.2.5 Evaluation

The usual method of evaluating classifiers is ten-fold cross-validation [42]. The set of labeled training examples is divided randomly into ten equally sized groups (folds) of examples. For each of these folds, we train the classifier on the other nine folds and then test its accuracy on the chosen fold. We take the average accuracy over all ten trials as an estimate of the performance of the classifier.

We believe that this method of evaluating classifiers is not wholly sufficient for evaluating genre classifiers. In traditional Text Classification tasks, the classifier being evaluated distinguishes between documents based on their topic. Thus in general, topic dependence is a desirable feature of a text classifier. However, genre spans several topics and thus we need to evaluate the genre classifier's performance across several subject domains. The tradition evaluation method of cross-validation gives no indication of how well the classifier will perform when trained on documents from one subject domain and tested on documents from another subject domain. We introduce a new complementary method of evaluation for genre classification that gives us a better estimate of the overall effectiveness of a classifier.

2.3 Text Classification

Automatic Text Classification is the task of automatically assigning a document to one of a set of categories [18]. Attributing a document to a category is done on the basis of the content of the document. Depending on the classification task, each document may be assigned a single category or a document may belong to several categories. Categorization may be one of two kinds: category-pivoted or document-pivoted. Document pivoted categorization occurs when documents are submitted one at a time for categorization and the categories are fixed in advance. This is the case in document filtering. Category pivoted categorization occurs when new categories can be added once a certain number of documents have been classified.

A typical application involves assigning documents to a pre-defined set of categories based on the topic of the document. An example of such an experiment is given in [17]. Data was collected from 20 newsgroups, 1000 documents collected from each. The task of the classifier was to automatically recognize which newsgroup a document came from. A naive Bayes classifier achieved an average accuracy of 89% on this task.

There has been a large amount of work in the application of Machine Learning techniques to Text Classification. We mention a representative sample. The CORA system [29] is a hierarchical catalog of computer science research papers. Documents are automatically spidered from the web and Machine Learning techniques are used to assign the document a place in the subject hierarchy. YahooPlanet [32] uses the Yahoo![1] hierarchy as a basis for automatic document categorization and applies Machine Learning techniques to the task of automatically assigning web documents to a category within this hierarchy. Chen and Dumais [7] describe a system that automatically classifies search results into an existing category structure, allowing the user to browse search results by category rather than as a single list. The category structure is topic based. We envisage that a similar system that allows the user to browse search results by genre would be useful.

Document filtering is the categorization of a dynamic stream of incoming documents. Documents that are considered relevant to the user are automatically identified and documents considered less relevant are either discarded or presented with a lower ranking. An example is filtering news article. Grou-

[1] http://www.yahoo.com

plens [38] is a system for filtering Usenet news articles. This filtering system uses collaborative filtering for recommendation of news articles. Usenet articles are recommended to users on the basis that other users with similar interests recommended them.

Webwatcher [2] is a browsing assistant that learns the users preferences and attempts to interactively recommend links to the user as they are browsing the web. It task is to identify the links on the current page that are most promising for the users current information need. Letizia [27] is another browsing assistant that tracks the users browsing behaviour and attempts to infer their preferences based on this behaviour.

Adeater [25] uses Machine Learning to filter out advertising images from web pages, based on, among other things, the text associated with the image tags. This allows users to filter out advertising when browsing the web. Sahami [40] applies similar techniques to the filtering of junk e-mail. A Machine Learning algorithm learns to classify e-mail messages as being either spam or legitimate e-mail, using features based on the occurrence of certain keywords in the message. EmailValet [28] investigates the use of various Machine Learning algorithms to route e-mail messages to various devices, depending on their level of importance.

2.3.1 Overview of current methods

Machine Learning (ML) can be used to automate the construction of text classifiers. A general inductive process automatically constructs a classifier by observing the characteristics of documents that have already been labeled by a human expert. The use of ML techniques reduces the time and effort that a human needs to spend building a classifier. Instead of painstakingly constructing rules by hand, the expert labels some example documents and let the Machine Learning algorithm take care of building the classifier. The amount of expertise required is reduced to that of being able to label documents rather than being able to construct a classifier.

Using ML for Text Classification relies heavily on standard techniques from information retrieval (IR) [41, 47]. IR techniques are used to extract features from the documents, which can be used by the ML algorithm. IR style indexing is usually applied to all documents. This includes building a model of the terms and their frequency of occurrence over the entire document collection. Documents are represented as a sparse vector of the terms that occur in the document collection. This is known as a bag-of-words (BOW) representation. Indexing of documents is performed on the full text of the document. Before indexing, some pre-processing may be done on the document. The removal of stop-words is almost always done. Stop-words are words that are regarded as topic neutral and have little discriminatory power.

Stemming [35] is sometimes performed, although its utility is unproven. Stemming is the conflation of different forms of the same word into a single token. Techniques for dimensionality reduction (DR) may be used to reduce the size of the feature vector. DR can be by term selection where the chosen terms are a subset of the original terms or term extraction where the chosen terms are obtained by transformations or combinations of the original terms. When performing DR, the aim is to keep the terms that provide the most information. One method for choosing which terms to discard is document frequency. Removing the most infrequently occurring terms doesn't affect the retrieval performance. Another technique is to remove terms that only occur in a small number of documents e.g. remove all terms that occur in less than 3 documents. Latent Semantic Indexing [12] reduces the number of terms by mapping terms that tend to co-occur together into the same region of the feature-space.

Although the BOW approach to feature selection is most common in Text Classification, other approached to representing documents have being tried. Instead of representing the document using the terms that occur in it, it can be represented using phrases, synonyms or hyponyms [43].

When classifying web pages, additional features that are specific to web pages can improve the performance of retrieval and Text Classification systems. For example, Google treats words that occur in the text of links as a good description of the web page that the link points to. The meta-tags of web pages can provide extra information about the page that can be used to improve Text Classification [34]. Efforts have also been made to construct a computer understandable knowledge base, the content of which mirrors that available on the web [11].

2.4 Personalization

The ultimate goal in information retrieval and filtering systems is accurate personalization. The volume of information currently available to users means that a 'one size fits all approach' in no longer sufficient. Information retrieval systems should be able to rank the information retrieved according to the preferences of the user. Document filtering systems should be able to exclude documents that are irrelevant to the user.

Current personalization systems use content-based recommendation [33], collaborative filtering [14] or both [44, 31]. We believe genre classification can add a new dimension for personalized recommendation systems. Documents can be retrieved or filtered based on the text genre of the document.

In the remainder of this chapter we focus on the task of automatic genre classification. We survey the related work in this area and discuss the different approaches to the meaning of genre, the features used to automatically identify genre and the approaches taken to automatically classifying genre. We relate these different approaches and representations to our own.

Whereas the preceding sections covered general background and related work in the areas of Machine Learning and Text Classification, the rest of this chapter focuses specifically on work directly related to automatic genre classification.

2.5 Genre classification

There is at present a small amount of research into genre classification itself. We mention here a number of papers that deal with issues particularly relevant to genre classification. The main points for comparison with our work are:

1. Definition: Our notion of the concept of a genre class is consistent with much of the related research, but there are some differences re-

garding what is meant by a genre class.

2. Genre Taxonomy: We identify two genre classes which we use in our experiments. Other approaches may identify a set of genre classes for study or they may use the classes that are given in a particular dataset.

3. Datasets and Algorithms: There are no datasets that are targeted towards genre classification. We created our own genre classification by hand classifying documents that were spidered from the web. The creation of a genre dataset involved significant time and effort. Several studies use standard text corpora such as the Brown corpus. However, these corpora were not created with genre classification in mind and contain no mark-up related to genre class. We use C4.5 as our learning algorithm.

4. Feature-Sets: We investigate the utility of three different feature-sets for genre classification. The feature-sets we investigate are bag-of-words, part-of-speech and a hand-crafted heuristic approach. These features are described in detail in chapter 3.

5. Evaluation: We propose an additional method of evaluating the performance of genre classifiers. Other work tends to use only standard methods of evaluation for genre classifiers. We measure domain transfer to estimate how well the classifier transfers to new subject domains.

In our work we have identified two sample genres for use in our experiments. These genres are subjective or objective news articles and positive or negative reviews. Wiebe [48] defines subjectivity classification as distinguishing sentences used to present opinions and evaluations from sentences used to objectively present factual information. She investigates subjectivity classification at the sentence level and concludes that the presence and type of adjectives in a sentence is indicative of whether the sentence is subjective or objective. We seek to perform subjectivity classification at the document level.

Tong [46] describes a system that focuses on tracking various entities and the opinions being expressed about them. The opinions are tracked by monitoring online public discussion forums. In particular, they monitor online discussion about movies and determine the level of buzz associated with specific movies as they move from announcement of release, through opening weekend and on to extended distribution. Opinions are extracted using sentiment models. These are patterns that capture the way people talk about movies and use a set of custom lexicons that cover personal emotions, movie features and language tone. These are also used to model the tone of the opinion. It appears that they use heuristics to identify positive and negative opinions being expressed about particular movies. We seek to automate the classification of reviews as positive or negative.

Stamatatos et al. [45] describe the need for robust Text Classification tools that can be easily ported to other domains and natural languages. Without explicitly mentioning domain transfer, they recognize the need for classifiers that can easily transfer to new domains. However they do not elaborate on the on evaluation of transfer. Their notion of genre is similar to ours.

They propose to use as style markers the most frequently occurring words of the entire written language and show that the frequency of occurrence of the most frequent punctuation marks contains very useful stylistic information that can enhance the performance of an automatic text genre classifier. In essence they propose using information that would be discarded using traditional methods as features i.e. stop-words.

They note that the Brown corpus, which is used for experiments in much of the genre classification literature, is not built for genre detection purposes and that texts included in it are not always stylistically homogeneous. Instead the Wall Street Journal corpus is used for experiments. They constructed a genre corpus from this corpus consisting of 4 press genres: editorial, letters to the editor, spot news and reportage. For the purpose of building the genre dataset, genres were identified using simple keyword phrases e.g. text with the phrase 'letter to the editor', were classed as belonging to the 'letters to the editor' genre.

Discriminant analysis (DA) [15] was used to detect the text genre. DA takes a set of pre-categorized training documents and extracts a set of discriminant functions that distinguish the documents. The features used are the most commonly occurring words. They find that the best error rate is achieved using the 30 most frequent words and performance does not improve beyond this threshold. Adding the 8 most common punctuation marks as features lowers the error rate significantly.

This approach is domain and language independent and requires minimal computation. They do not perform any experiments to measure the performance of their classifier when it is transferred to new subject domains.

This work is closely related to ours. They identify the need for domain transfer but do not develop this idea any further. Their definition of text genre is similar to ours and two of the genre classes they identify are similar to our subjectivity classification task. The features they use are stop-words and punctuation. Some of our hand-coded features use stop-words and punctuation.

Roussinov et al. [39] view genre as a group of documents with similar form, topic or purpose, "a distinctive type of communicative action, characterized by a socially recognized communicative purpose and common aspects of form". This is a more general view of genre where genre is a grouping of similar documents. Some genres are defined in terms of purpose or function, others in terms of physical form while most documents combine the two.

They attempt to identify genres that web users frequently face and propose a collection of genres that are better suited for certain types of information need. To this end, they performed a user survey to see

1. what is the purpose for which users search the web, and
2. whether there was a relation between the purpose of a respondents search and the genre of document retrieved.

This survey showed that purposes included scholarly research, shopping, health, news. They assigned genres to search results and measure agreement between genres assigned by interviewers and respondents. For example, for scholarly research useful genres include table of contents, articles, essays, bibliographical records, books, tutorial pages, slides. They attempt to identify 5-6 groups of web genre that satisfy the following two properties:

1)relationship of each page to each group can be established with accuracy of 80-100%, and 2) each group is likely to satisfy a limited subset of observed search purposes.

They propose a set of genres along with a set of features for each genre. There are no experimental results to back up any of these proposals, they are based on intuition. For example, a group called publications containing genres scholarly research, news, financial. They also propose a user interface for genre based searching. This study is at the early stages with no implementation or experiments. The main contribution is the identification through user interviews of genres that can usefully assist in improving retrieval of web documents.

Dewdnew et al. [13] take the view that genre of a document is the format style. Genre is defined as a "label which denotes a set of conventions in the way in which information is presented". The conventions cover both formatting and the style of language used. They differentiate between information filtering and sorting by genre because the former carries out text selection based on content. They use two feature-sets: a set based on words (traditional BOW) and a set based on the way in which the text is presented. They use the maximum entropy characterization of information gain to pick word features. Presentation features include linguistic features such as prevalence of adjectives, use of tense, sentence complexity and layout features such as line spacing, tabulation and non-alphanumeric characters. The dataset contained 7 genres: advertisement, bulletin board, FAQ, Message board, radio news, reuters newswire, television news. Three algorithms were used: naive Bayes, C4.5 and Support Vector Machines [18]. Analyzing word gain over the whole corpus resulted in 323 word based features being selected. Over all the corpus, C4.5 achieves average recall of 79.8% for word frequency features and 85.3% for presentation features. Combining the feature-sets gives a slight improvement. they conclude that linguistic and format features alone can be used successfully for sorting documents into different genres. It is interesting to note that their presentation features do consistently better than the word frequency features.

Rauber and Muller-Koller [37] present a way to provide automatic analysis of structure of text documents. They argue that in a traditional library, non content-based information such as age of a document and whether it looks frequently used are important distinguishing features. They present a way of providing automatic analysis of the structure of documents. This analysis is based on various surface level features of the text such as word statistics, punctuation information, occurrences of various special characters and keywords, mark-up tags capturing various image, hyperlink, equation and similar information. The approach uses a self-organizing map (SOM) [24] to cluster the documents according to structural similarities. They describe a graphical interface to the SOMLib digital library system. This depicts documents as hard-cover or paperback books, binders or papers sorted by content into various bookshelves. Documents are colored according to their structural similarities. So complex descriptions stand apart from summary documents on the same topic. Document content is represented in a manner similar to our BOW approach. They want their classification process to be as language and domain independent as possible to allow flexible application of the system in different settings. Several different types of features are identified. Text complexity features are based on word statistics such as number of words per sentence, average word length, number of sentences, number of paragraphs. Other features include special characters and punctuation counts (e.g. ! is indicative of emotional text, ? indicative of interviews), certain stop-words (e.g. I, you, probably, highly) and mark-up tags (these indicate information such as the number of images present in the document, tables, equations etc.). Some of these features are features that we use for our hand-coded feature-set.

A content based SOM provides an organization of documents by their subject, while a genre SOM analyses the structural features of the documents and groups the documents accordingly. The genre map is visualized by mapping it onto an RGB color-space so that documents that are of similar genre are also of similar color.

In this approach the genres are identified from clusters of documents that occur in the SOM. In our approach we identify the genre classes first and then build classifiers to recognize them. A combination of content-based and genre-based techniques give the user more power to discriminate between documents.

Kessler et al.[22] argue that genre detection based on surface cues is as successful as detection based on deeper structural properties. In general genre refers to the kind of text used. They refer to attributes used in classifying genres as generic facets and generic cues are the observable properties of a text that are associated with facets. Types of cues include structural cues (such as POS tags, passives, nominalizations, lexical cues), character level cues (such as punctuation and separators), derivative cues (such as ratios and variation measures derived from lexical and character level features). Structural cues have the disadvantage of requiring tagged or parsed texts so experiments were based on 55 cues from the last 3 groups using texts from the Brown corpus. The genre classes were based on using a small number of categories from the Brown corpus that could be considered genres. The genre categories were reportage, editorial, sci-tech, legal, non-fiction and fiction. They used 499 texts with a training corpus of 402 and a test corpus of 97. They did not use cross-validation. Logistic regression and neural network were used as learning algorithms. For the reportage and editorial classes a baseline majority classifier scores 81%, LR scores 89%for reportage and 75% for editorial. Thus their attempts to automatically identify subjective documents performed worse than the baseline.

Karlgren [19, 20, 21] has done several experiments in genre classification. In [19] he defines a genre as a consistent and distinguishable tendency to make certain linguistic choices and aims to find methods to complement topical information retrieval. The corpus is based on the Wall Street Journal. As features, they use word-based statistics (such as word length, long word counts), text based statistics (such as sentence length, sentence complexity) and other statistics (such as pronoun counts, presence of contradictions or amplifiers, frequency of verbs such as 'seem', 'appear'). They regard a genre as a grouping of documents that are stylistically consistent and intuitive to accomplished readers of the channel in question. The genre palette depends on the document collection in question. A set of TREC queries with the WSJ corpus were used for evaluation. It was found that the texts that were judged relevant did differ systematically (in terms of style) from the texts that were not relevant.

In [20], Karlgren et al. use topical clustering in conjunction with stylistics based genre prediction to build an interactive information retrieval engine and to facilitate multi-dimensional presentation of search results. They claim that traditional methods of IR based on content are shallow and sparse. A text has more facets and features than a list of terms can represent. Documents are more complex than is assumed by traditional search methods: a document is more than term frequencies and more than the topic that it is about. They define a genre as a set of documents with a perceived tendency to make the same stylistic choices. They built a genre palette by interviewing users, aiming to find genres that are reasonably consistent with what users expect and are observable and computable using measures of stylistic variation. They identified 11 main genre classes that are useful for web filtering. These are

1. informal, private, personal home pages;
2. public commercial;
3. interactive pages such as feedback forms and searchable indexes;
4. journalistic materials such as news, editorials, reviews;
5. scientific, legal, formal text;
6. other running text;
7. FAQs;
8. Link collections ;

9. other listings and tables;

10. Discussions; and

11. Error messages.

A variety of features are used: lexical features (such as relative frequency of personal pronouns, emphatic expressions), textual and genre specific features (such as average word length, relative number of digits) and internet specific features (such as number of images, number of HREF tags). They use C4.5 as their learning algorithm to build the classifier. Evaluation was through user trials. Twelve users were given two search tasks. The subjects did not do well on the search tasks, but all but one reported that they liked the genre enhanced search interface. Subjects used the genres in the search interface to filter the search results, however the classifier makes mistakes too often for some genres.

The search interface described is an example of how genre classification can usefully aid information retrieval. There is no attempt to evaluate domain transfer of genre classifiers. The genre palette identified for the web seems promising although some of the proposed genres may need to be divided into sub-genres to be useful for document filtering.

Argamon et al. [1] consider two types of features: lexical and pseudo syntactic. Lexical features are simply the frequency of function words such as about, and, the. For their second approach, they tag every word in the text using Brill's POS tagger. They then consider POS trigrams as features in the hope that these trigrams encode useful syntactic information. They use ripper [9] as their learning algorithm. The classifier was tested on 4 collections: New York Times news, New York Times editorial, New York daily news and Newsweek. They tested the ability to distinguish between each of 6 category pairs. e.g. for NY times editorial vs. NY times news, using function words as features achieved accuracy of 78%, while using POS trigrams achieved accuracy of 69%. For NY times editorial vs. Newsweek, function words achieved accuracy of 61% while POS trigrams achieved accuracy of 70%.

Biber's study "variation across speech and writing" is the major work in the area of stylistics [3]. Biber investigates the difference between speech and written text across several text genres. He compares spoken and written texts along dimensions of linguistic variation. These dimensions define continuums of variation rather than discrete poles. Examples include formal/informal, literary/colloquial, restricted/elaborated.

He examines the variation of 67 linguistic features in 481 texts. His aim is to identify groups of features that actually co-occur in texts. The linguistic dimension is given priority over the functional dimension. This is in contrast to the approach taken in other research where a dimension is identified e.g. formal/informal and then features are identified that can be used to classify that dimension.

He uses the term genre to refer to text categorizations made on the basis of the external criteria related to author speaker purpose. He adopts his genre categories from the distinctions used in the corpora. This gives 15 genres: press reportage, editorial, press review, religion, popular lore, skills and hobbies, biographies and essays, official documents, academic prose, fiction (general, mystery, science, adventure, romantic) and humor.

He aims to systematically described the linguistic characteristics of the range of genres in English whether spoken or written. He attempts to identify the linguistic parameters on which genres vary so that any genre can be located within an oral and literate space which specifies the difference and similarities between the genre and range of other genres in English.

There are 67 linguistic features investigated in the study . Frequency counts for all linguistic features are normalized. He identifies features that tend to co-occur together. He examines six dimensions and the features that have discriminatory power along that dimension:

1. Involved vs. informational production
2. Narrative vs. non narrative concerns
3. Explicit vs. situation dependent reference
4. Overt expression of persuasion
5. Abstract vs. non Abstract information
6. Online informational elaboration

and identifies co-occurrence patterns among linguistic features associated with each dimension. He then compares each of the genre categories which respect to these dimensions.

For example, editorials score higher on dimension six while press reportage and reviews score low. He distinguishes genres from text types. Genres characterize text on the basis of external criteria, while text types represents groupings of texts that are similar in linguistic form. He also investigates variation within sub genres. In the press reportage genre, some types of press reportage are more colloquial, affective and involved than other types. Cultural press reportage is most involved while financial is the least involved. Spot news has a remarkably high narrative focus while financial reportage is markedly non narrative. Editorials are relatively homogeneous, but sub genres in this category show interesting differences along dimension 4. Letters to the editor are more persuasive than personal editorials. Biber's works is not predictive but rather analyzes variation in texts. This corpus is likely fairly homogeneous in comparison to our web corpus.

Summary

Much of the related work uses some features that we investigate in our study. However, most use hybrid feature-sets that use some POS features and some word-based features. They don't seek to investigate the difference in performance between different types of features. Some mention the need for genre classifiers that transfer well to new domains but don't investigate what features can provide this transfer or how to evaluate it.

The next chapter described in detail the different features we used to build our classifiers.

Chapter 3

Feature Selection

In order to apply machine learning techniques to genre classification tasks we must supply a set of examples for the training algorithm. The learning algorithm then uses these examples as a basis for building a model with which future predictions can be made. Since the examples are documents, the learning process requires that a set of features be identified and extracted from these documents to be used as input for the learning algorithm.

Input takes the form of instances and attributes. The concept being described is the genre class we are attempting to automatically classify. In the case of the subjectivity classification task, the concept is whether a document is subjective or objective. The other concept we consider is whether a review is positive or negative.

The instances correspond to the set of training examples used to build the classifier model. Each training example corresponds to a single sample instance of the concept being described.

Each instance is characterized by a set of attributes. Each attribute measures a different aspect of the particular instance. The set of attributes are pre-defined and must be identified prior to the learning process. The choice of features to be used as attributes can affect the quality of the model generated by the learning algorithm. It is important to identify the salient features. We are interested in identifying features that perform well on the genre classification task and transfer well to new subject domains.

Preparing the input to the learning algorithm

It is the preparation of the input to the learning algorithm that takes the majority of time and effort in the learning process. First a set of training examples must be collected. In both our genre classification tasks, the training examples were gathered from the WWW.

In order to run experiments, the set of training examples must be manually classified according to which concept class they belong to. In the subjectivity classification task, the training instances were all hand-classified. This involved a significant effort and when building any classifier it is undesirable to have to hand-classify the training examples. In the case of the positive or negative concept, the training examples were classified automatically, by downloading the examples from review sites. We wrote a set of wrappers to automatically extract the rating of a particular review and split the ratings as being either positive or negative.

We must then identify a set of features which we will use to represent each instance and encode the training instances using these attributes. We investigate different feature-sets and the quality of model that can be generated using these feature-sets. We are interested in investigating which features are relevant to the genre classification tasks and which features maintain their relevance across multiple subject domains.

The first set of features are extracted using a standard bag of words (BOW) approach. This approach is widely used in text classification.

The second feature-sets is a part-of-speech (POS) based approach. Documents are represented by their POS tags rather than the actual words occurring in them.

Both these feature-sets have the advantage that they can be automatically extracted from a document collection and are independent of the particular genre classification task being considered.

As a comparison to the performance of these feature-sets, we constructed a third feature-set for the subjectivity classification task. This feature-set was based on a set of heuristics. These heuristics were identified by examining the datasets in the football and politics domains and by asking people for their intuition on what features would be good for this genre classification task.

Identifying and extracting this feature-set involved significantly more effort than the first two feature-sets. The aim of using this feature-set was to attempt to measure the performance of a classifier built using features identifies by a human as being likely to perform well on this task. We wish to compare how the two automatically extractable generic feature-sets perform with respect to the domain specific feature-set which was built with human assistance.

Once we have identified our features, we must extract the features from the data collection and encode them in a format suitable for use as input to the learning algorithm. We performed our experiments using the Waitico environment for knowledge analysis (WEKA)[1].

3.1 Keywords

The first approach represented each document as a bag-of-words. In the BOW representation, a document is encoded as a feature-vector, with each element in the vector indicating the presence or absence of a word in the document. This approach was used as a baseline to determine how well a standard keyword based learner performs on this task. This approach led to feature-vectors that are large and sparse.

This approach to document representation is a standard approach in text classification. Documents are represented as a large vector of binary features. Each feature represents the presence or absence of a particular word that occurs in the document collection, in the document. The BOW approach is described in section 2.3.1. In our experiments we used stemming and removed stop-words. We used the stop-word list provided with the BOW toolkit [30].

The procedure for encoding our labeled dataset is simple. First we build a text model for the entire dataset. This involves extracting every unique word in the dataset. We then merge words in the document than have the same stem. e.g. "compute" and "computing" would count as the same word. Stemming is done using the Porter stemming algorithm [35]. We remove stop-words from the text model using a list of stop-words. This gives us a base

[1] http://www.cs.waikato.ac.nz/~ml/weka/

Document 1:

```
This file contains stuff about computing and
computers.
```

Document 2:

```
This second sample file contains information
about computers, biology and other stuff.
```

Figure 3.1: Two example documents

	biologi	comput	file	inform	sampl	stuff	subjectivity class
D1	0	1	1	0	0	1	objective
D2	1	1	1	1	1	1	objective

Table 3.1: An example of the BOW encoding process

vector for the entire document collection. All documents in the collection can be represented as a similar vector with each element of the vector indicating the presence or absence of a word in the document. This base vector also gives us our feature-set for input to the learning algorithm. We use all the features in the vector plus the actual class of the document to represent the instance. This typically gives vectors of around 10,000 features for our document collections.

Figure 3.1 shows the contents of two short examples. Table 3.1 gives an example of the BOW encoding process on these examples. It shows the BOW feature representation of the two examples.

This BOW feature-vector approach is common in text classification. It works well for standard text classification where the aim is to categorize documents according to subject or topic. This is because certain words are strongly indicative of a particular topic. They are much more likely to occur in one class of documents than the other.

It is not obvious whether certain keywords would be indicative of the genre of the document. We are interested in investigating how well this standard text classification approach works on the genre classification tasks. We expect that a classifier built using this feature-set may perform well in a single subject domain, but not very well when domain transfer is evaluated. By subject domain we mean a group of documents that can be regarded as being about the same general subject or topic. For example, for the subjectivity classification task, we have three subject domains: football, politics and finance. For the review classification task we have two subject domains: restaurant reviews and movie reviews. The reason we identify different subject domains is that a text genre class may occur across multiple subject domains. We wish to evaluate the domain transfer of a genre classifier. For example, if a classifier is trained for the subjectivity classification task using documents from the football domain, how well does it perform when this classifier is transferred to the new domain of politics?

In the normal use of the BOW approach, the target of classification is the topic of the document. In these cases this approach works well. In the case of genre classification however, the target concept is often independent of the topic of the document, so this approach may not perform as well. This feature-set has the advantage of being easy to automatically extract from a document collection. However it results in very large, sparse feature-vectors. It can be computationally expensive to build classifiers using this approach.

3.2 Part-of-Speech statistics

The second approach uses the output of Brill's part-of-speech (POS) tagger [4] as the basis for its features. It was anticipated that the POS statistics would reflect the style of the language sufficiently for our learning algorithm to distinguish between different genre classes. A document is represented as a vector of 36 POS features, one for each POS tag, expressed as a percentage of the total number of words for the document.

This approach uses a part-of-speech representation of the documents rather than the actual words occurring in the document. It was hoped that this would give a representation that was still capable of discriminating genre but was independent of the subject of the document. The POS representation does not reflect the topic of the document, but rather the type of text used in the document.

We tag each word in the document with its appropriate POS tag. The POS tag explains how the word is used rather than what the word is. POS tags indicate whether a word is a noun, verb, pronoun, adjective, preposition, conjunction or interjection. These are the basic parts of speech. A word can occur in a document as different part-of-speech, depending on how it is used. For example, 'I bought a book'. Here the word 'book' occurs as a noun. However, in the sentence 'I will book the tickets', it is a verb. We expect the POS of a document are informative with respect to the genre class of the document.

We use Eric Brill's POS tagger, a trainable rule-based tagger. We use the rule-set arising from training the tagger on the Brown corpus. The tagger

	CC	CD	DT	EX	subjectivity class
D1	0.125	0	0.125	0	objective
D2	0.087	0	0.087	0	objective

Table 3.2: An example of a subset of POS encoding

uses the Penn Treebank tagging style and tag-set. Input to the tagger must follow the Penn Treebank style, i.e. one sentence per line, punctuation split off from words etc. The Penn Treebank tokenization is discussed in more detail in section A.1.1. The Penn Treebank tag-set defines a set of 36 POS tags which expand the basic POS tags to give more detailed information. Section A.1.2 lists the tags with examples.

For the purpose of representing our documents using this feature-set, we tag each word in the document with its POS tag. We then count the number of occurrences of each POS tag in the document. This gives us 36 numerical features. We normalize each of these features by dividing by the number of words in the document. Figure 3.2 shows an example encoded with a subset of the POS features on the example files from figure 3.1. For display reasons, the example shown uses only four POS tags to represent a document. The full POS representation uses 36 tags to represent a document.

We hope that POS features can be used to differentiate genres in a domain independent way. Wiebe[48] indicates that the number of adjectives in a sentence can indicate whether the sentence is subjective or objective. We hope that this extends to document level for the subjectivity task. If the POS feature-set is capable of differentiating genre class, we would expect that it would do so in a domain independent manner as it doesn't have any

	ave_sen_len	very	long words	n_I	n_!	subjectivity class
D1	8	0	3	0	0	objective
D2	12	0	6	0	0	objective

Table 3.3: An example of a subset of HC encoding

information about the topic of the document.

As with the BOW approach, this approach can easily be applied automatically to a document collection. Also it can be computationally expensive for large documents. It results in a small feature-set though.

3.3 Hand-Crafted feature-set

Our third approach is to use a set of hand-crafted features. Some of these are features mentioned in related work that intuitively seem suited to the task of subjectivity classification e.g. having lot's of sentences beginning with a capital 'I', while others were selected from examination of the training data for the football and politics domains. Many of the features could be used for other genre identification tasks but some are specific to the task of identifying subjective documents. In total there were 76 features. Examples include counts of certain stop-words, counts of various punctuation symbols, and text statistics such as average sentence length, number of long words and keywords that indicate subjectivity within the domain.

This feature-set was an attempt to build an optimal classifier for the subjectivity classification task. i.e. how well can we do when a person armed with domain knowledge about the classification task attempts to design a custom feature-set for that classification task.

The task was to design a set of features that intuitively seem useful for the task of subjectivity classification. The features were selected from several sources. Some were identified by examination of the football and politics domains for the subjectivity classification task. Others were selected from related work. We also asked people to read some sample documents and suggest features for the subjectivity classification task.

The full feature-set is described in appendix A.2. Some sample features include the average sentence length, the number of sentences beginning with 'I' and the occurrence of the word 'opinion' in the document.

The identification of features for this feature-set was manual and thus involved significantly more effort than the previous two datasets. It gives a good base for comparison for the other two feature-sets. This HC feature-set was designed with the genre classification task of subjectivity classification in mind, and thus can be regarded as a reasonable estimate of the best performance we can achieve with lots of time, effort and domain knowledge. If the other feature-sets perform to a similar level as this one then we prefer to use those feature-sets as they are easy to automate across different genre classification tasks and across multiple subject domains for any particular genre class.

This set of features is dependent on the genre classification in question. It takes a lot of effort to identify a feature-set and then the feature-set is likely to be useful for just one genre classification task. Because of this, the other two feature-sets would be preferable if they can achieve similar levels of performance.

	automatically extractable	expected to be topic independent	expected to be genre independent
BOW	yes	no	yes
POS	yes	yes	yes
HC	no	yes	no

Table 3.4: Summary of expected performance of feature-sets

3.4 Summary

Table 3.4 summarizes the expected performance of the three different feature-sets which we aim to investigate. Both the Bow approach and POS approach can be easily automatically extracted from any document collection. The HC approach however, requires a human to identify what features are to be used and to write a program to automate the extraction of those features.

We expect that the POS approach will be topic independent while the BOW approach will be topic dependent. The POS approach represents documents in a way that doesn't encode the topic of the document, but represents the type of text used in the documents. The BOW representation of a document is tied to the topic of that document and thus we would not expect it to be topic independent. Whether the HC approach is topic independent depends on the features that are chosen. In the case of the features we have identified to build our HC feature-set, we would expect it to be topic independent as most of the features are not topic related, but consist of stop-words and text statistics.

We expect that both the POS and BOW approach will be genre independent as both can easily be applied to any particular genre task. The HC approach cannot easily be applied to a new genre task as features must be identified for that task.

In the next chapter, we perform experiments on two particular genre classification tasks to verify whether these feature-sets perform as expected on the genre classification task.

Chapter 4

Experiments

4.1 Introduction

In this chapter we evaluate our methods for genre classification. We perform experiments on two genre classification tasks. The first task is that of subjectivity classification; classifying articles according to whether they convey the opinions of their authors. The second genre classification task is classifying reviews as being either positive or negative.

For each genre classification task we wish to investigate several things. We wish to evaluate the performance of the classifier using three feature-sets: BOW, POS and HC as described in chapter 3. We evaluate the performance of classifiers in two ways. We evaluate performance within a single subject domain using the standard technique of ten-fold cross-validation. We also introduce a new evaluation criterion: domain transfer, with which we seek to evaluate the performance of the classifier across multiple subject domains.

The classifier we use for these experiments is C4.5. We performed a preliminary set of experiments using several learning algorithms on the subjectivity classification task. We present these experiments to justify our choice of classifier.

We investigate the learning rate of C4.5 on the subjectivity classification task using each feature-set. We present learning curves for each feature-set in each subject domain and investigate whether there is a learning rate advantage for any of our feature-sets.

For each of the two genre classification tasks, we describe in detail the experiments performed. We describe the dataset and how it was collected. We present the results of the various experiments and analyze these results.

Our experiments show that all three feature-sets can successfully be used for the subjectivity classification task. When the classifier is evaluated in a single subject domain, the hand-crafted features perform best, while the POS features perform worst. However when we evaluate domain transfer in addition, POS is the best performer. We conclude that the POS feature-set is best for this task.

For the review classification experiment, the BOW features perform well in a single subject domain. POS performs poorly on this classification task. Neither approach performs well when domain transfer is evaluated. We conclude that this is a more difficult classification task and that it is difficult to build a classifier for this task that will transfer well to new domains.

4.2 Aims

Our aim is to develop classifiers that have high accuracy but also have good domain transfer. Classifiers that have high accuracy but poor domain transfer are of limited use. Because a genre can occur across multiple subject domains, we want our genre classifier to perform well on subject domains that it was not trained on.

The effort to develop a classifier is considerable, we hope that the classifiers developed will transfer well to new subject domains. To this end, we hope to use our domain transfer experiments to identify features which lead to general domain independent classifiers, while preserving high accuracy.

We wish to investigate the performance of different feature-sets on genre classification tasks and which features are good for building general domain independent classifiers.

We are interested in what features can be used to indicate genre. How much of genre information is conveyed by the topic of the document and how much is conveyed by the style of text used in the document.

We evaluate our classifiers using several measures, accuracy of the classifier in a single subject domain, accuracy when trained on one subject domain but tested on another and domain transfer rate which indicates the classifier's ability to perform well in new subject domains.

We wish to build classifiers that have high accuracy in a single subject domain. In addition to this we wish to build classifiers with a good domain transfer rate.

Figure 4.1 shows single domain accuracy plotted against transfer rate,

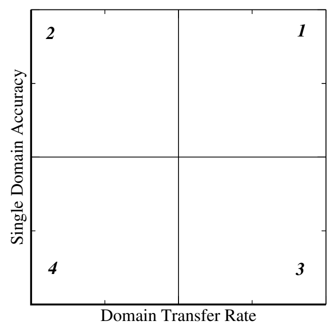

Figure 4.1: Desirable performance characteristics of a genre classifier

with areas of the graph labeled in order of desirability. The most desirable classifiers would be in region 1 of the graph. These classifiers have both high single domain accuracy and high transfer rate. Next most desirable are classifiers occurring in region 2 of the graph. These classifiers have high single domain accuracy but poor domain transfer. Classifiers occurring in regions 3 and 4 are undesirable because they have poor levels of single domain accuracy and even if they have good transfer rates, they are not useful in practice.

4.3 Evaluation Methods

The standard method of evaluating Machine Learning classifiers is to use cross validation. We believe that for the task of genre classification, this alone is not sufficient. An extra method of evaluation is needed. Because genre classes are at least partially orthogonal to topic we need to measure the classifiers performance across topic domains as well as across genre classes. We use standard cross validation to measure accuracy[1] within a single subject domain. We also propose a domain transfer measure to evaluate the

[1] Other measures could be used such as precision, recall or F-measure

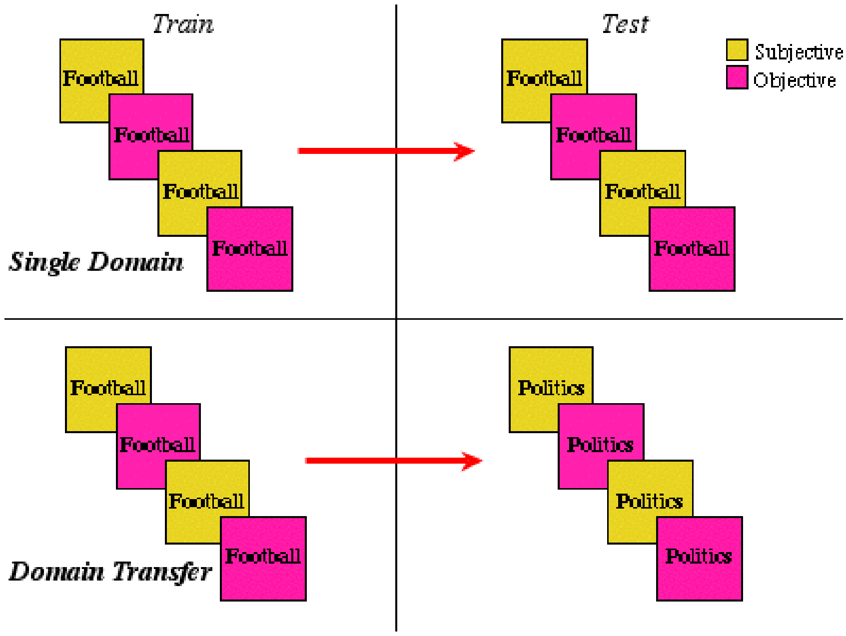

Figure 4.2: Domain Transfer

classifier's ability to generalize across topic domains.

Figure 4.2 shows what we mean by domain transfer. In the single domain case, the classifier is trained and tested on documents from the same subject domain. In the domain transfer case, the classifier is trained on documents from one subject domain and tested on documents from another subject domain.

Usually text classification is applied to tasks where topic specific rules are an advantage. However since genre is orthogonal to topic this is not the case for genre classification. In the case of genre classification, topic specific rules reduce the generality of the genre classifier. In addition to evaluating the genre classifier's performance in a single topic domain, we also need to evaluate its performance across multiple topic domains.

4.3.1 Single Domain Accuracy

Single domain accuracy measures the accuracy of the classifier when it is trained and tested on instances from the same subject domain. This measure indicates the classifier's ability to learn the classification task in the subject domain at hand.

Accuracy is defined as the percentage of the classifier's predictions that are actually correct as measured against the known classes of the test examples. Accuracy is measured using ten-fold cross-validation. This involves dividing the dataset randomly into 10 equal parts. In turn each of these parts is used as a test set, while the others are used as training sets. Accuracy is measured in each case and averaged across all 10 trials.

This is a standard technique used in Machine Learning to evaluate classifiers. However the normal use of Machine Learning techniques in text classification is to classify documents according to topic or subject. Thus the ability to identify topic specific features is an advantage in most applications of text classification. Tenfold cross validation is a suitable method for evaluating these classifiers. The genre of a document is independent of the topic of the document. We aim to build genre classifiers that are general and perform well on unseen documents. Traditional evaluation techniques give us no indication of how well our genre classifier will perform on documents from other subject domains. We introduce a new evaluation measure, domain transfer, which indicates the classifier's performance on documents from other subject domains.

4.3.2 Domain Transfer Accuracy

We measure domain transfer in an attempt to measure the classifier's ability to generalize to new domains. For example, a genre classifier built using documents about football should be able to recognize documents about politics from the same genre. We use the domain transfer measure as an indicator of the classifier's generality. It also gives us an indication of how much the genre classification task in question is topic dependent or topic independent.

Domain transfer is evaluated by training the classifier on one subject domain and testing it on another subject domain.

We use this domain transfer accuracy as a indication of the classifier's performance on documents from other subject domains. We also measure the average performance of the classifier in the domain transfer situation with respect to the performance in single subject domains by calculating the domain transfer rate.

Domain Transfer Rate

In addition to measuring the domain transfer accuracy, we calculate the domain transfer rate. This measures how much the classifier's performance degrades when the classifier is evaluated on new subject domains. A classifier that performs equally well in the transfer condition as in a single domain would achieve a transfer score of 1. A classifier whose performance degrades when transferred to new subject domains would achieve a transfer score of less than 1.

Consider a classification task consisting of a learning algorithm C and a set of features F. Let $D_1, D_2, ..., D_n$ be a set of subject domains. Let ${}_DA_D$ be the performance of C when evaluated using ten-fold cross-validation in domain D. Let ${}_{D_1}A_{D_2}$denote the performance of classification scheme C when trained in domain D_1 and tested in domain D_2 . We will use accuracy as our measure of performance. We define the domain transfer rate for classification scheme C as

$$ {}_CDT_F = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1;j\neq i}^{n} \left(\frac{{}_{D_i}A_{D_j}}{{}_{D_i}A_{D_i}} \right) \tag{4.1} $$

We evaluate the quality of a genre classifier using both single domain accuracy, domain transfer accuracy and the domain transfer rate. Ideally we would hope to get high single domain accuracy and a high domain transfer rate. A classifier with high accuracy and low transfer may be useful in some situations but not in others.

4.4 Motivation for choosing C4.5 for experiments

We chose to use C4.5 as our learning algorithm. This decision was taken for a number of reasons. We performed initial experiments using several different learning algorithms (OneR, Naive Bayes, k-nearest neighbor and C4.5) to determine whether any particular learning algorithm was likely to be more suited to the genre classification task. This initial experiment is similar to that described in section 4.6. The task was to classify documents as being subjective or objective. We performed experiments using four learning algorithms: OneR, Naive Bayes, K-nearest neighbor and C4.5. These algo-

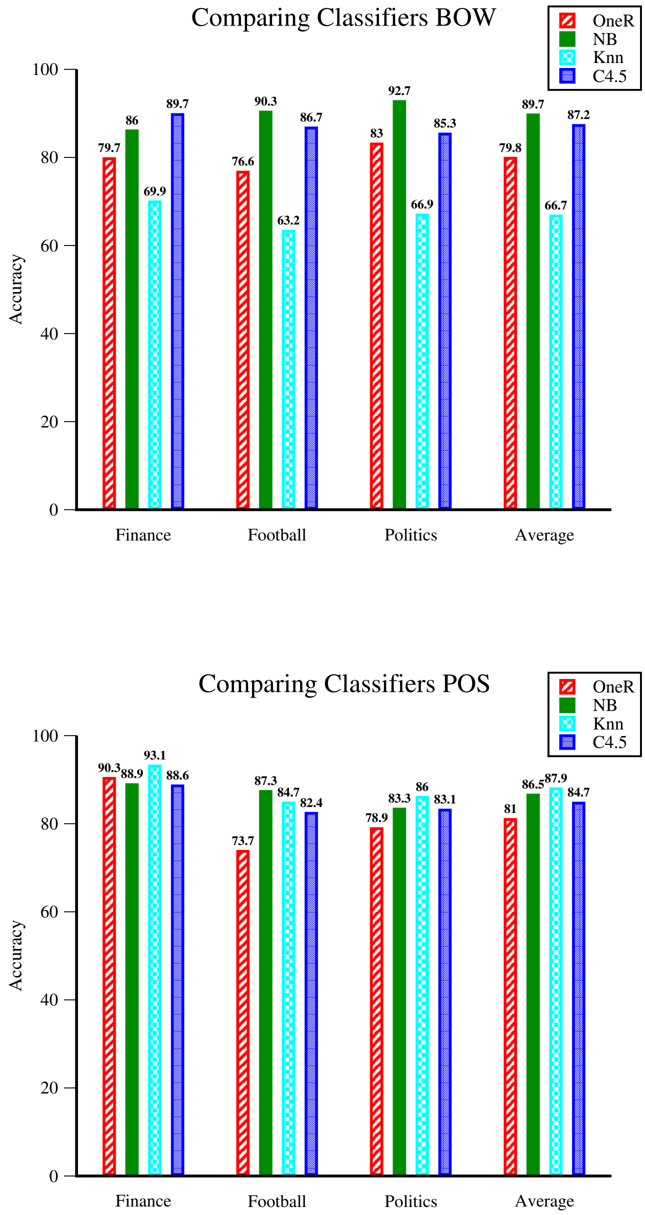

Figure 4.3: A comparison of different learning algorithms on the subjectivity classification task

rithms are described in chapter 2. We investigated the performance of these algorithms using two feature-sets: bag-of-words and part-of-speech. The aim was to determine if any particular learning algorithm performed significantly better on this task.

Figure 4.3 shows the results of these preliminary experiments. The graphs shows accuracy for the subjectivity classification task across three subject domains; football, politics and finance. The vertical axes shows accuracy as measured within a single subject domain. Experiments were performed using two feature-sets: BOW and POS.

KNN[2] performs best on average using the POS features(87.9%) but worst on average using the BOW features. This is probably due to the size of the feature-space in each case. POS has a small number of features(36) while the BOW approach results in large feature-vectors (approximately 10,000 features). The large number of features and the sparsity of occurrence of those features may reduce the effectiveness of the KNN technique. The difference in the performance of KNN between the two feature-sets makes KNN an unsuitable algorithm for our experiments as we wish to investigate differences in feature-sets rather than differences that occur as a result of the learning algorithm. The consistence of performance of NB and C4.5 across the two feature-sets indicate that the difference here is as result of the learning algorithm.

Naive Bayes and C4.5 perform similarly well on both feature-sets. NB performs slightly better in both cases, but the difference is small, 2.5% for BOW and 1.8% for POS. On average there is a small difference in performance be-

[2] Experiments used a value of k=5

tween C4.5 and NB. If we look at the subject domains individually, in the BOW approach C4.5 outperforms NB in one of the three domains(finance), while in the POS approach C4.5 and NB perform almost identically in two of the three subject domains (finance and politics).

OneR performs surprisingly well in comparison to C4.5 when its simplicity is taken into account. C4.5 generates complex decision trees while OneR generates simple one level decision trees. In the BOW approach OneR achieves an average accuracy of 79.8% compared to 87.2% for C4.5, a difference of 7.4%. For the POS approach OneR achieves accuracy of 81% compared with 84.7% for C4.5, a difference of only 3.7%. It is surprising that OneR performed so well on this genre classification task as it is such a simple classification algorithm. This is consistent however with Holte's [16] findings when comparing C4.5 and OneR on some of the UCI datasets.

There is very little difference in performance between NB and C4.5 although NB gives slightly better performance. However there are reasons other than performance to use C4.5:

- comprehensibility and confidence: a person can examine the generated model to understand the learning task. Decision trees generated by C4.5 are easy to understand and can be interrogated by a human user. Users can gain confidence in the model generated by the classifier by examining it. The model generated by naive Bayes is more difficult for a human to understand.

- C4.5 is widely regarded as a good learning algorithm that has been shown to achieve good performance on a wide variety of datasets.

Subject Domain	Opinion	Fact	Total
Football	174	177	351
Politics	144	145	289
Finance	56	100	156

Table 4.1: Corpus Details for the Subjectivity Classification Experiment

- It is an efficient algorithm in terms of speed of execution.
- Decision trees can easily be converted into a set of rules. This makes it easy to implement a classifier as a set of rules.

4.5 Experimental Setup and Document Corpora

Table 4.1 shows the details of the corpus used for the subjectivity classification experiment. We identified three subject domains, football, politics and finance. For each of these topics, we identified a number of news web sites that specialize in news in the particular subject domain. Articles were then automatically spidered from these sites over period of several weeks. The documents were then classified by hand by the author as being either subjective or objective.

Table 4.2 shows the details of our document collection for the review experiment. The documents were gathered automatically from the web. The collection of review datasets was somewhat easier than the collection of the subjectivity datasets. The reason is that the classification of a document could be extracted automatically using a wrapper for the particular site. For example, most movie reviews come with a recommendation mark. A review that awards a film 4 stars could be considered a positive review, while a review that awards a film 1 star could be considered a negative review. Thus we automatically extract the classification of a particular review, negating the need to manually classify each document.

The Movie reviews were downloaded from the Movie Review Query Engine (MRQE)[3]. This site is a search engine for movie reviews. It extracts movie reviews from a wide range of sites. If the review contains a mark for the film, the mark is also extracted. We wrote a wrapper to extract a large number of movie reviews and their corresponding marks from this site. Because MRQE extracts reviews from many different sites, many of them have different marking schemes. For example, some have a maximum mark of 4 stars, others 5 stars, while others use an A,B,C,D scheme. The marks from various sites were normalized by converting them to a percentage and then we used documents with high percentages as examples of positive reviews and vice versa. Marks below 41 were considered negative while marks of 100% were considered positive.

The restaurant reviews were gathered from the Zagat survey site[4]. This is a site that hosts a survey of restaurants from the U.S.A. and Europe. Users of the site submit their comments about a particular restaurant and assign marks in three categories (food, decor and service). The marks for these categories are between 1 and 30 and are the average for all the users that have provided feedback on that particular restaurant. The reviews themselves consist of an amalgamation of different users comments about the restaurant. We averaged the marks for the three categories to get a mark for each

[3]http://www.mrqe.com
[4]http://www.zagat.com

Subject Domain	Positive	Negative	Total
Movie	386	337	723
Restaurant	300	331	631

Table 4.2: Corpus Details for the Review classification experiment

restaurant. Restaurants that got an average mark below 15 were considered negative while those getting marks above 23 were considered positive.

4.6 Experiment 1: Subjectivity Classification

The first experiment was on the genre classification task of subjectivity classification. The task was to classify documents as being either subjective or objective i.e. does the document report news objectively or does it convey the opinion of the author. Table 1.1 shows examples of extracts from subjective and objective documents. The reader can discern subtle differences in the tone and style of the text for each class. We seek to automate the classification of documents as being either objective or subjective. We use the feature-sets described in chapter 3 and the learning algorithm C4.5.

4.6.1 Results

Figure 4.4 shows the results of our single domain evaluation for the subjectivity classification experiment. Figures 4.5 and 4.6 show the domain transfer evaluation on the subjectivity classification task, excluding and including the finance domains respectively. Figure 4.7 shows the results when we train the classifier on two subject domains and test it on the third subject domain. In each of the graphs, the vertical axis shows the accuracy of the classifier.

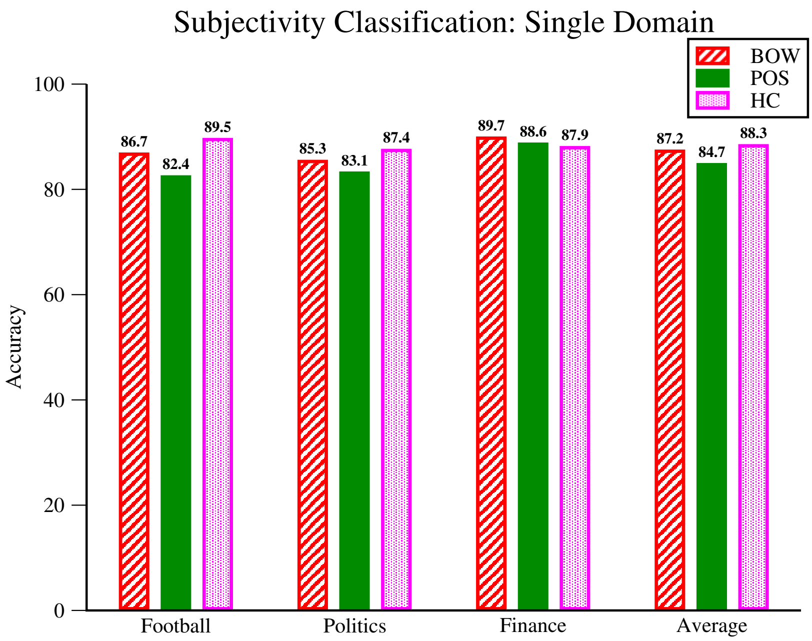

Figure 4.4: Subjectivity classification within a single domain

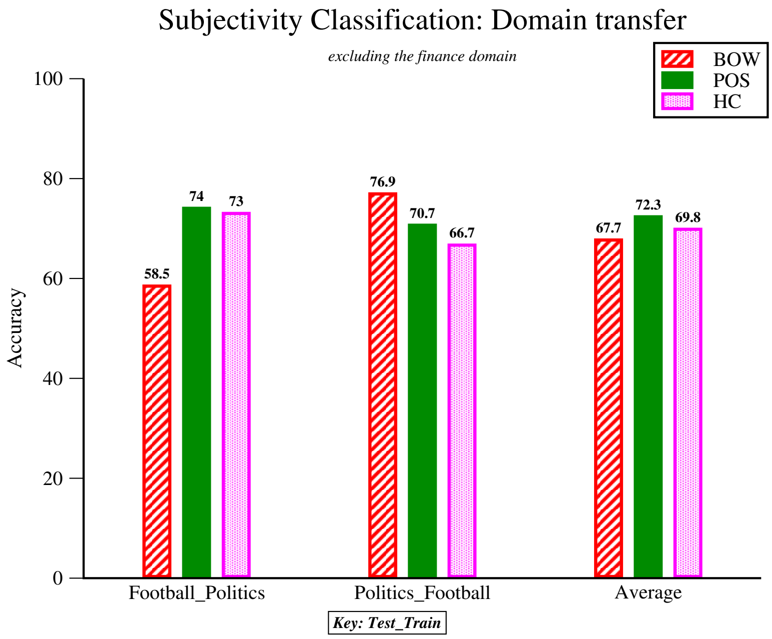

Figure 4.5: Domain transfer for the subjectivity classification task excluding the finance domain

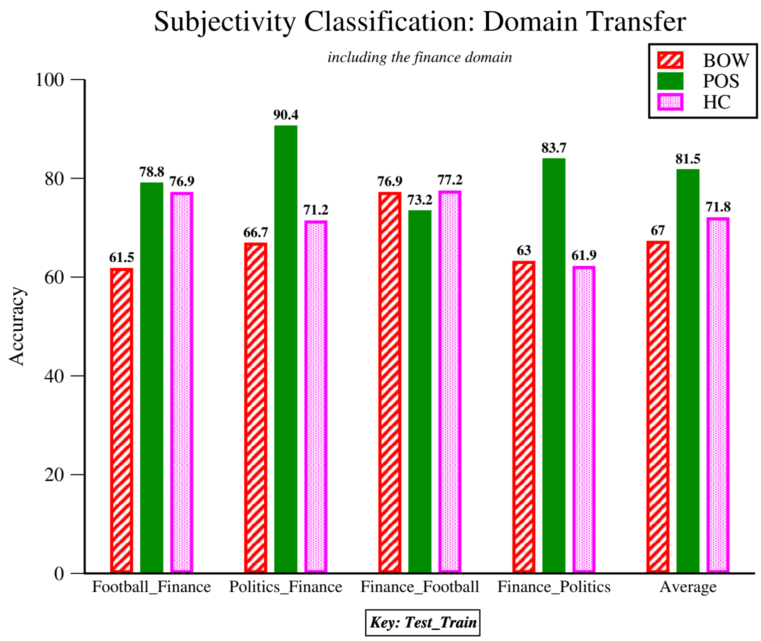

Figure 4.6: Domain transfer for the subjectivity classification task including the Finance Domain

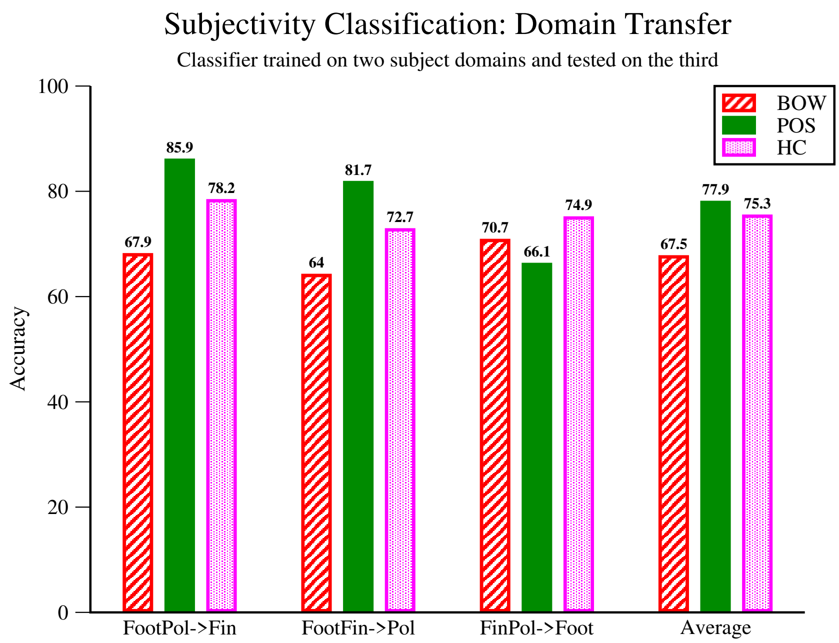

Figure 4.7: Domain transfer for the subjectivity classification task when training on two of the subject domains and testing on the other

4.6.2 Analysis

Single Domain Experiments

In the single domain experiments (Figure 4.4), the hand-crafted features perform best with average accuracy of 88%. POS statistics performs worst (85%), while BOW achieves average accuracy of 87%. On average there is little difference in accuracy between the three classifiers when evaluated in a single domain. This indicates that subjectivity within a single domain can be accurately identified using any of the three representations.

Note that on average the hand-crafted features perform best but are actually worst in the finance domain. This is consistent with the fact that some of the hand-crafted features were designed by examination of the Football and Politics datasets. We conclude that while it is usually easy to come up with a feature-set that works well, crafting a set of features that generalize well to other domains is more difficult.

Domain Transfer Experiments

Since our hand-crafted feature-set was designed by examination of the football and politics datasets, we examine cases including the finance domain separately.

Figure 4.5 shows domain transfer results excluding the finance domain. On average POS performs best (72%), HC is second (70%), while BOW is worst (68%). For experiments that use the finance dataset (shown in Figure 4.6) this pattern is repeated, but the differences are more significant. POS is again best (82%), HC is second (72%) and BOW is last (67%).

In the domain transfer experiments, the POS features perform significantly better. The BOW features perform worst. This indicates that while keywords can be used to identify subjective documents, a model built using these features is more closely tied to the document collection used for training. Intuitively we would expect that the classifier built using the POS statistics as features would have a more generalizable model of what constitutes genre than one built using keywords or domain-specific hand-crafted features.

The classifier built using hand-crafted text statistics performs better than the keyword classifier but worse than the POS classifier. Shallow text statistics have the advantage of being easy and inexpensive to compute. We believe that many of these features would be useful for identifying genres other than our opinion/fact task. Chapter 2 gives describes previous work in the area of genre classification, some of which uses features that are included in our hand-crafted feature-set.

Note that, as predicted the HC features perform worse than the POS features when a third domain is considered. This is not unexpected as the HC features were designed through examination of the football and politics domains, but not the finance domains. Thus, performance of the HC features is poorer in the finance domain.

The BOW approach performs similarly to the other two approaches in single domain experiments but is inferior when domain transfer is evaluated. POS statistics and shallow text statistics are both good feature-sets for this particular genre identification task. Hand-crafted text statistics perform best in single domain experiments, while POS statistics perform best in domain transfer experiments.

It must be remembered that the hand-crafted features were designed with our classification task in mind and after examination of some of the training data from two of the domains. Therefore the hand-crafted features have an inherent advantage for this particular classification task. We expect that the utility of each technique depends on the particular genre at hand and that a combination of the feature-sets might be suitable for a general genre classifier.

POS statistics and BOW based features can be generated automatically. However, the selection of hand-crafted features requires human involvement and these features may not transfer well to a new domain. When building a classifier to identify a different genre class some of the hand-crafted features used here would be useful. Other features would need to be gleaned from examination of the target genre.

Figure 4.7 shows domain transfer from an experiment in which we trained the classifier on two of the three subject domains and tested it on the third. On average, the POS approach performs best followed by HC with BOW performing worst. Training in two domains rather than one does not significantly improve the performance of the resulting classifiers. In the HC case, the classifiers trained on two domains perform better on average than those trained on a single domain. For the BOW and POS approaches, the classifiers trained on two domains perform to approximately the same level.

Domain Transfer Rate

We can calculate the domain transfer rate described in section 4.3.2. Expanding equation 4.1 gives us

$$ {}_{C}DT_{F} = \frac{1}{6}\left(\frac{{}_{fb}A_{pol}}{{}_{fb}A_{fb}} + \frac{{}_{fb}A_{fin}}{{}_{fb}A_{fb}} + \frac{{}_{pol}A_{fb}}{{}_{pol}A_{pol}} + \frac{{}_{pol}A_{fin}}{{}_{pol}A_{pol}} + \frac{{}_{fin}A_{fb}}{{}_{fin}A_{fin}} + \frac{{}_{fin}A_{pol}}{{}_{fin}A_{fin}}\right) $$

$$ {}_{C4.5}DT_{BOW} = \frac{1}{6}\left(\frac{58.5}{86.7} + \frac{61.5}{86.7} + \frac{76.9}{85.3} + \frac{66.7}{85.3} + \frac{76.9}{89.7} + \frac{63}{89.7}\right) = 0.771 $$

$$ {}_{C4.5}DT_{POS} = \frac{1}{6}\left(\frac{74}{82.4} + \frac{78.8}{82.4} + \frac{70.7}{83.1} + \frac{90.4}{83.1} + \frac{73.2}{88.6} + \frac{83.7}{88.6}\right) = 0.93 $$

$$ {}_{C4.5}DT_{HC} = \frac{1}{6}\left(\frac{73}{89.5} + \frac{76.9}{89.5} + \frac{66.7}{87.4} + \frac{71.2}{87.4} + \frac{77.2}{87.9} + \frac{61.9}{87.9}\right) = 0.81 $$

The ordering of this domain transfer rate is similar to the observed domain transfer accuracies. i.e. POS performs best, followed by HC, with BOW performing worst. The differences in transfer rate between the three feature-sets are statistically significant[5]. POS performs statistically significantly better than both HC and BOW. HC performs significantly better than BOW.

[5] Statistical significance is measured using a paired t-test. $p < 0.05$ is regarded as significant

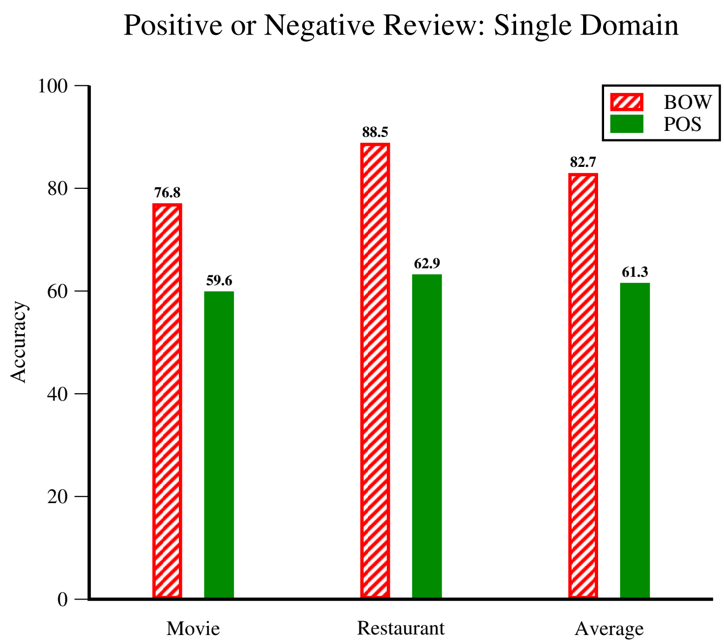

Figure 4.8: Review classification within a single domain

4.7 Experiment 2: Review Classification

4.7.1 Results

Figure 4.8 shows the results for the review experiments in a single domain. Figure 4.9 shows the domain transfer results for the review experiments. In each case, the results are shown for each domain along with the average performance over the two domains. The vertical axis lists the accuracy, as estimated using ten-fold cross validation. The results are shown for the BOW and POS approaches.

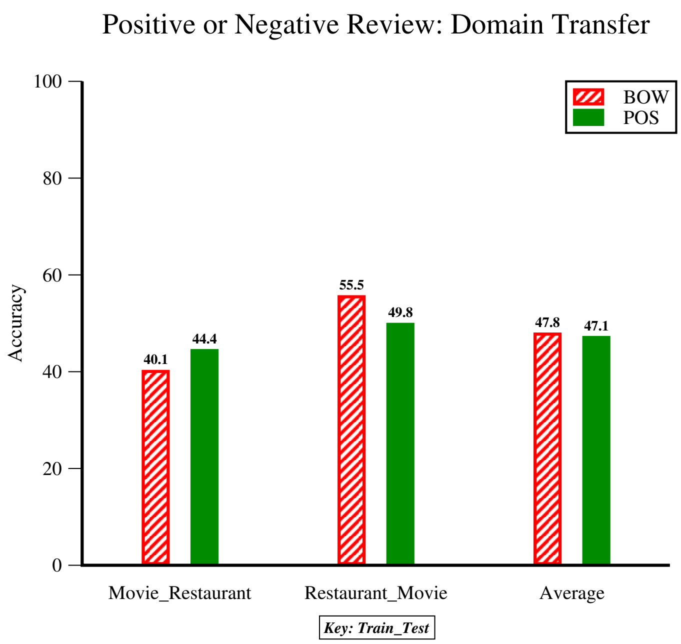

Figure 4.9: Domain transfer for review classification

4.7.2 Analysis

Single Domain Experiments

Figure 4.8 shows the single domain results for the review classification experiment. In both domains, the BOW approach performs significantly better than the POS approach. On average, the BOW approach achieves accuracy of 82.7%. This is a good level of accuracy for this classification task. The POS approach performs poorly in comparison(61.3% on average).

The BOW approach is capable of achieving good levels of performance when attempting to classify reviews as positive or negative in a single subject domain. The POS approach performs poorly on this classification task, even in a single subject domain.

Domain Transfer experiments

Figure 4.9 shows the domain transfer results for the review classification experiment. On average, both approaches perform to a similar level with there being less than 1% between them. In each domain, the classifiers perform poorly with both approaches. BOW achieves accuracy of 40% and 56%, while POS achieves accuracies of 44% and 50%. This level of performance is no better than achievable by a simple majority classifier.

The single domain experiment on this classification task showed that BOW can achieve high levels of accuracy on this classification task in a single subject domain. However the domain transfer experiment shows that the BOW approach fails when the transfer approach is evaluated. The BOW features which indicate a positive movie review are not transferable to the restaurant domain and vice versa. The POS approach fails in both the single domain and domain transfer experiments.

We conclude that the POS approach is not suitable for the task of classifying reviews as being either positive or negative. The BOW approach can achieve good performance in a single subject domain but cannot transfer to new subject domains. Even though the traditional means of evaluating a classifier indicate that the BOW achieves good performance, our experiments indicate that it performs poorly when we our extra domain transfer condition is evaluated.

Domain Transfer Rate

We can calculate the domain transfer rate described in section 4.3.2. Expanding equation 4.1 gives us

$$DT_C = \frac{1}{2}\left(\frac{{}_{mov}A_{rest}}{{}_{mov}A_{mov}} + \frac{{}_{rest}A_{mov}}{{}_{rest}A_{rest}}\right)$$

$${}_{C4.5}DT_{BOW} = \frac{1}{2}\left(\frac{40.1}{76.8} + \frac{55.5}{88.5}\right) = 0.575$$

$${}_{C4.5}DT_{POS} = \frac{1}{2}\left(\frac{44.4}{59.6} + \frac{49.8}{62.9}\right) = 0.77$$

The domain transfer rate for BOW is poor showing that it does not transfer well. For the POS approach, the original single domain accuracy was poor so the high domain transfer rate here is misleading. A high transfer rate is useless if the original single domain accuracy is low. Domain transfer rate must be considered in conjunction with the original single domain accuracy.

4.8 Experiment 3: Learning Curves

Figure 4.10 shows learning curves for the subjectivity classification task over the three subject domains. The vertical axis shoes the average accuracy with a particular proportion of the dataset used for training. The horizontal axis shows the proportion of the dataset used for training. Each point on the curve is the average of 10 runs with a particular proportion of the dataset

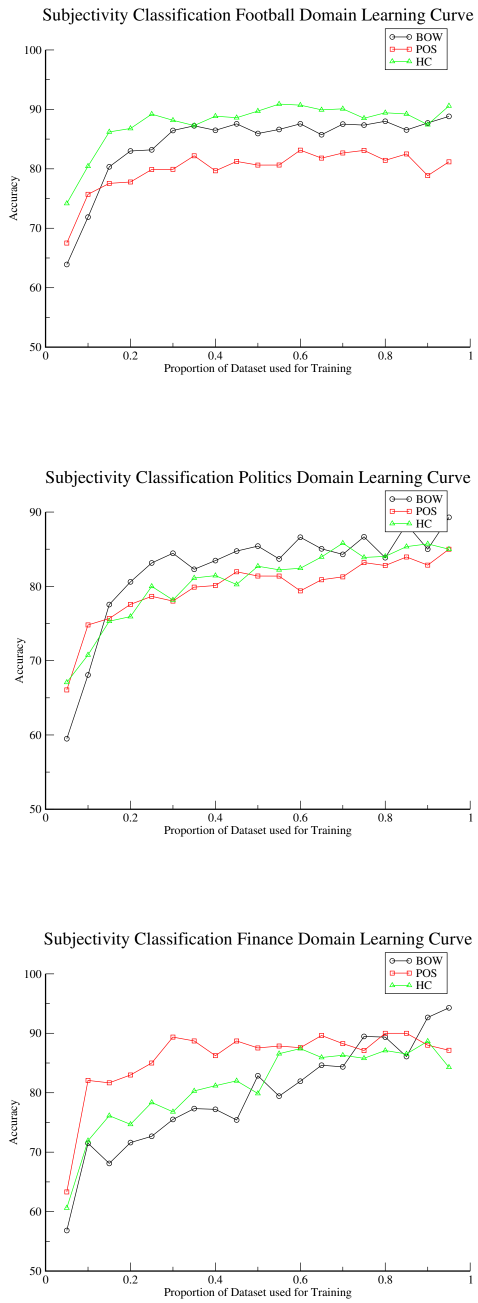

Figure 4.10: Learning Curves for the football, politics and finance domains on the subjectivity classification task

used for training and the remainder of the dataset used for testing.

The curve was generated as follows:

1. Randomly pick 5% of the dataset for training and evaluate the resulting classifier on the remaining instances.

2. Repeat this 10 times and take the average as the accuracy at that point on the curve.

3. Repeat steps 1 and 2 using {10,15,...95}% of the dataset for training.

We generated learning curves for the subjectivity classification task in each of the three subject domains. The first graph shows the learning curve in the football domain. In this case HC performs best, while BOW performs worst. The shape of the curve is similar for all three feature-sets, so the difference is in accuracy rather than learning rate.

The second graph shows the learning curve for the Politics domain. In this case BOW performs best, while there is little difference between POS and HC. BOW also has the better learning rate as it performs significantly better with small proportions of training data.

The third graph shows the learning curve for the finance domain. In this case, POS performs significantly better than the other two approaches. It has a better learning rate and reaches its maximum accuracy with only 35% of the dataset used for training.

In each case, the classifier reaches accuracy of 80% with small amounts of training data (20% of the dataset or less). Using more training data beyond this point results in much smaller gains in accuracy.

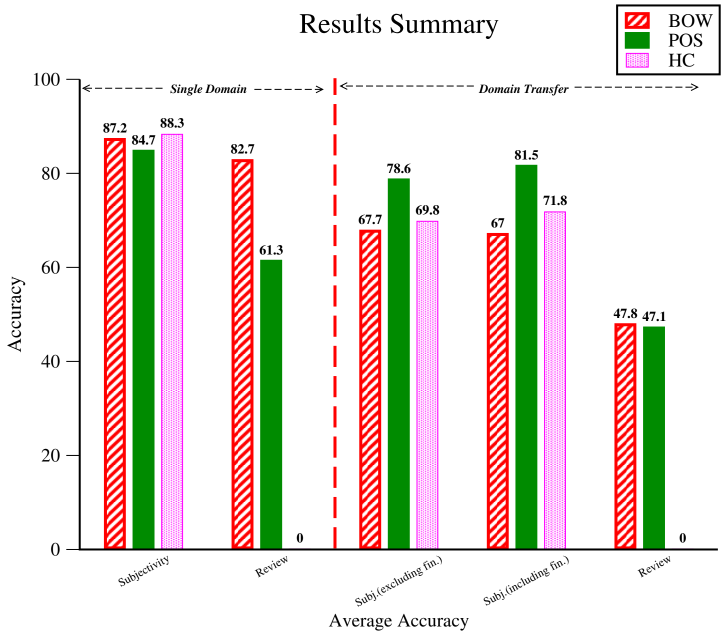

Figure 4.11: Summary of results

These learning curves do not show any of the approaches being better from the perspective of learning rate. They do show however that the classifiers can achieve good performance with reasonably small amounts of training data, and that subsequently increasing the size of the training set does not achieve significant increases in accuracy.

4.9 Summary of Results

Figure 4.11 shows a summary of the experimental results. The graph shows average accuracies for the single domain and domain transfer experiments. In the single domain experiments BOW performs better than POS. For the subjectivity experiment the difference between the two approaches is small, but in the review experiment the POS approach performs poorly in comparison

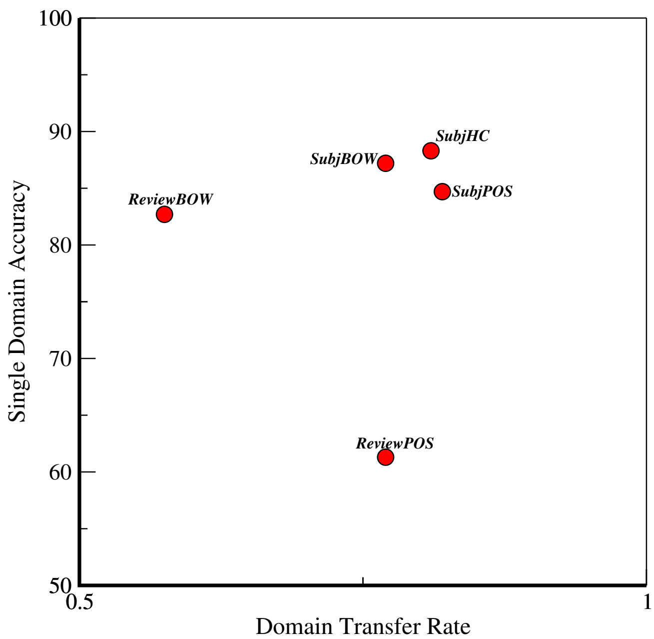

Figure 4.12: Summary of feature-set performance

to BOW. For the subjectivity experiment HC performs best of all.

For domain transfer experiments, the subjectivity results are divided into those including and excluding the finance domains. In each case, the POS features perform best while BOW performs worst. POS performs significantly better than either approach in the transfer situation. Neither approach performs well in the transfer situation for the review experiments. In fact neither approach performs any better than a random choice classifier would in the transfer condition.

Figure 4.12 graphs single domain accuracy against domain transfer rate. It shows that we achieved significantly better results on the subjectivity classification task than on the review classification task. For the subjectivity classification, POS and HC approaches are more desirable than the BOW approach. On the review classification task, BOW is a good approach for single subject domain classifiers, while the POS approach is not appropriate to this classification task.

We conclude that for the subjectivity classification task the POS approach is best. It achieves good accuracy in both the single domain and domain transfer experiments. For the task of classifying reviews as being either positive or negative the BOW approach is better as it performs well in the single domain case. Neither approach achieves any domain transfer.

We conclude that genre classification is feasible but challenging. It is possible to achieve high accuracy for particular genre classification tasks within a particular subject domain. however, when we evaluate these genre classifiers for domain transfer we find their performance can drop significantly. It is difficult to design genre classifiers that are general enough to perform well in new unseen subject domains.

Chapter 5

Conclusion

5.1 Overview

We have investigated the following issues in this thesis:

- Can genre be automatically identified using Machine Learning techniques?
- What features are useful for automatically identifying genre?
- How much are the genre classifiers that we build topic independent and how well to they transfer to new unseen subject domains?

More specifically, we have experimentally answered the following questions:

- Can news articles be automatically classified as subjective or objective?
- Can reviews be automatically classified as positive or negative?
- Which feature-sets perform best on these tasks?
- Which feature-sets yield classifiers with good domain transfer?

5.2 Discussion

In theory, genre and topic are orthogonal. However, our experiments indicate that in practice they partially overlap. It may be possible to automatically identify genre in a topic independent way, but the results of our domain transfer experiments show that the feature-sets we investigate result in models that are partially topic dependent.

From a single domain point of view, our approach was very successful. If we used only the usual methods of evaluation, we would conclude that genre classification is not a difficult task and can easily be achieved using standard machine learning techniques. On the subjectivity classification task, all our feature-sets achieved high accuracy, while on the review classification task a standard bag-of-words approach achieved good accuracy.

We have argued that standard methods of evaluation are not sufficient when evaluating genre classifiers and that in addition the genre classifiers ability to transfer to new topic domains must also be evaluated. When we evaluate this additional aspect of the genre classifiers, we find that it is difficult to build classifiers that transfer well to new domains.

We investigated two genre classification tasks and conclude the following for each:

For the subjectivity classification task:

- It is possible to build a genre classifier that can automatically recognize a document as being either subjective or objective.
- High accuracy in a single subject domain can be achieved using any of the three feature-sets we investigated (BOW, POS or HC)

- When domain transfer is measured for this task, the POS feature-set performs best.

- Overall, the POS feature-set is best for this genre classification task as it performs well both in a single subject domain and when transferred to new subject domains.

For the review classification task:

- It is possible to build a genre classifier that can automatically recognize a review as being either positive or negative. However, this is a more difficult task than the subjectivity classification task.

- Good accuracy can be achieved in a single topic domain using the BOW approach. The POS approach is not suitable for this genre classification task.

- Neither the BOW or POS approach achieve good domain transfer on this task.

- This is a difficult classification task and the best approach is to use a standard BOW feature-set as it performs well in a single subject domain.

From these we make the following overall conclusions:

- Genre Classification is challenging but feasible.

- We can build genre classifiers that perform well within a single subject domain, but building classifiers that transfer well to multiple subject domains is difficult.

- For the subjectivity classification task, the POS approach is best. On this task, this feature-set will result in a classifier that performs well in a single subject domain and transfers well to new domains.

- For the review classification task, the BOW approach is best. On this task, this feature-set performs well in a single subject domain, but does not transfer to other domains.

- When evaluating genre classifiers, it is important to consider their ability to transfer to new subject domains. Evaluating a genre classifier in a single subject domain can give a misleading indication of the overall usefulness of the classifier.

Table 3.4 showed the expected performance of the three different feature-sets. We expected the BOW approach to work for different genres but to be topic dependent. Our experiments support this. The BOW approach performs well on both genre classification tasks when evaluated in a single topic domain, but it performs poorly when evaluated across multiple topic domains. We expected the POS approach to be both topic and genre independent. This was supported by the subjectivity experiment, but not by the review classification experiment. For subjectivity classification, the POS approach performs well both in a single domain and transfers well to new domains. For review classification experiment, this approach does not perform well for single or multiple subject domains.

5.3 Contributions

We have made the following contributions:

- We investigated the feasibility of genre classification and the use of Machine Learning techniques for automatic genre classification.
- We investigated different feature-sets for the construction of genre classifiers and their usefulness for building genre classifiers that perform well both within a single subject domain and across multiple subject domains.
- We identified domain transfer as an important aspect of building useful genre classifiers and evaluated domain transfer for three feature-sets.

5.4 Future Work

We believe that genre classification has application in several areas. These include information retrieval, text classification, document filtering systems and personalization. Currently genre classification has not been widely used in any of these areas.

Indeed the techniques for automatically classifying documents according to genre have not been thoroughly investigated. In this work, we have provided a starting point for the construction of automatic genre classifiers. However the issues involved in building such automatic genre classifiers need to be further investigated.

We identified two sample genre classification tasks. These particular genre classes could be usefully applied to improve existing systems. In section 1.4

we described possible applications of these genre classes. Applications that utilize genre classification to provide noticeable benefits to the end user must be developed to illustrate that genre classification can be a useful, practical technique that can be used to improve the results of document retrieval systems.

In building such systems it will be useful to identify additional genres that can improve a users ability to filter documents and reduce the number of documents that are potentially relevant to them. An expanded genre taxonomy is needed together with appropriate techniques for automatically identifying genres. We found that the techniques that were successful on one genre classification task (subjectivity classification), were less successful on another genre classification task (review classification).

The ability to achieve good domain transfer is important for genre classifiers. The techniques we used did not provide a complete separation of genre and topic. Further investigation is needed to determine methods of identifying genre in a topic independent way.

Ideally once a general genre taxonomy is defined we need techniques for automatically constructing genre classifiers within this taxonomy. One would hope that there are general techniques that could be used to build all classifiers for all genres within a taxonomy and that these genre classifiers will transfer easily to new subject domains. However, our experience has shown that this is difficult and methods for achieving these aims need further investigation.

In general future work consists of extending the work we have done on two genre classification tasks to a general genre taxonomy. Classifiers built to identify genre classifiers within this genre taxonomy should be easy to build and domain independent. The other major area for future work is to implement applications that use genre classification to improve user experience.

Bibliography

[1] Shlomo Argamon, Moshe Koppel, and Galit Avneri. Routing documents according to style. In *First International Workshop on Innovative Information Systems*, 1998.

[2] Robert Armstrong, Dayne Freitag, Thorsten Joachims, and Tom Mitchell. Webwatcher: A learning apprentice for the world wide web. In *AAAI Spring Symposium on Information Gathering from Heterogeneous Distributed Environments*, 1995.

[3] Douglas Biber. *Variation across speech and writing.* Cambridge University Press, 1988.

[4] Eric Brill. Some advances in transformation-based parts of speech tagging. In *AAAI*, 1994.

[5] Sergey Brin and Lawrence Page. The pagerank citation ranking: Bringing order to the web. http://www.db.stanford.edu/ backrub/pageranksub.ps.

[6] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[7] Hao Chen and Susan T. Dumais. Bringing order to the Web: automatically categorizing search results. In *Proceedings of CHI-00, ACM International Conference on Human Factors in Computing Systems*, pages 145–152, Den Haag, NL, 2000. ACM Press, New York, US.

[8] George R. Clare. *The Measurement of Readability.* Iowa University Press, 1963.

[9] William W. Cohen and Yoram Singer. A simple, fast, and effective rule learner. *AAAI*, 1999.

[10] Scott Cost and Steven Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10:57–78, 1993.

[11] Mark Craven, Dan DiPasquo, Dayne Freitag, and Andrew McCallum. Learning to extract symbolic knowledge from the world wide web. In *AAAI*, 1998.

[12] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George M. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[13] Nigel Dewdney, Carol VanEss-Dykema, and Richard McMillan. The form is the substance: Classification of genres in text. In *ACL Workshop on Human Language Technology and Knowledge Management*, 2001.

[14] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *CACM*, 35(12):61–70, 1992.

[15] Trevor Hastie, Robert Tibshirani, and Andreas Buja. Flexible discriminant analysis by optimal scoring. *Journal of the American Statistical Association*, 89(428):1255–1270, 1994.

[16] R. C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–91, 1993.

[17] Thorsten Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In Douglas H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 143–151, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.

[18] Thorsten Joachims. A statistical learning model of text classification with support vector machines. In W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel, editors, *Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval*, pages 128–136, New Orleans, US, 2001. ACM Press, New York, US.

[19] J. Karlgren. Stylistic experiments in information retrieval. In T. Strzalkowski, editor, *Natural Language Information Retrieval*. Kluwer, 1999.

[20] Jussi Karlgren, Ivan Bretan, Johan Dewe, Anders Hallberg, and Niklas Wolkert. Iterative information retrieval using fast clustering and usage-specific genres. In *Eight DELOS workshop on User Interfaces in Digital Libraries*, pages 85–92, Stockholm, Sweden, 1998.

[21] Jussi Karlgren and Douglass Cutting. Recognizing text genres with simple metrics using discriminant analysis. In *Proceedings of the 15th. International Conference on Computational Linguistics (*COLING 94*)*, volume II, pages 1071–1075, Kyoto, Japan, 1994.

[22] Brett Kessler, Geoffrey Nunberg, and Hinrich Schutze. Automatic detection of text genre. In *ACL/EACL*, 1997.

[23] Jon Kleinberg. Authoritative sources in a hyperlinked environment. In *Nineth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[24] Teuvo Kohonen. The self-organising map. *Proceedings of IEEE*, 78(9):1464–1479, 1990.

[25] Nicholas Kushmerick. Learning to remove internet advertisements. In *3rd Int. Conf. on Autonomous Agents*, 1999.

[26] David D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine*

Learning, number 1398, pages 4–15, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[27] Henry Lieberman. Letizia: An agent that assists web browsing. In *CAI*, 1995.

[28] Sofus A. Macskassy, Haym Hirsh, and Aynur A. Dayanik. Emailvalet: Learning email preferences for wireless platforms.

[29] Andrew McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. A machine learning approach to building domain-specific search engines. In *IJCAI*, 1999.

[30] Andrew Kachites McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. 1996. http://www.cs.cmu.edu/ mccallum/bow.

[31] P. Melville, R. Mooney, and R. Nagarajan. Content-boosted collaborative filtering. *SIGIR Workshop on Recommender Systems*, 2001.

[32] Dunja Mladenić. Turning YAHOO! into an automatic Web page classifier. In Henri Prade, editor, *Proceedings of ECAI-98, 13th European Conference on Artificial Intelligence*, pages 473–474, Brighton, UK, 1998. John Wiley and Sons, Chichester, UK.

[33] Raymond J. Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. In *Proceedings of DL-00, 5th ACM Conference on Digital Libraries*, pages 195–204, San Antonio, US, 2000. ACM Press, New York, US.

[34] John M. Pierre. On the automated classification of web sites. In *Linkoping Electronic Articles in Computer and Information Science*, volume 6. 2001.

[35] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[36] Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, 1993.

[37] A. Rauber and A. Muller-Kogler. Integrating automatic genre analysis into digital libraries. In *First ACM-IEEE Joint Conf on Digital Libraries*, 2001.

[38] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, , and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the Conference on Computer Supported Collaborative Work*. Transcending Boundaries, 1994.

[39] Dmitri Roussinov, Kevin Crosswell, Mike Nilan, Barbara Kwasnik, Jin Cai, and Xiaoyong Liu. Genre based navigation of the web. In *34th International Conference on System Sciences*, 2001.

[40] Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. A bayesian approach to filtering junk E-mail. In *Learning for Text Categorization: Papers from the 1998 Workshop*, Madison, Wisconsin, 1998. AAAI Technical Report WS-98-05.

[41] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[42] Steven Salzberg. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1(3):317–328, 1997.

[43] Sam Scott and Stan Matwin. Feature engineering for text classification. In Ivan Bratko and Saso Dzeroski, editors, *Proceedings of ICML-99, 16th International Conference on Machine Learning*, pages 379–388, Bled, SL, 1999. Morgan Kaufmann Publishers, San Francisco, US.

[44] Barry Smyth and Paul Cotter. Surfing the digital wave: Generating personalised TV listings using collaborative, case-based recommendation. *Third International Conference on Case-Based Reasoning*, 1999.

[45] E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Text genre detection using common word frequencies. In *18th International Conference on computational Linguistics*, 2000.

[46] Richard M. Tong. An operational system for detecting and tracking opinions in on-line discussions. In *SIGIR Workshop on Operational Text Classification Systems*, 2001.

[47] C. J. van Rijsbergen. *Information Retrieval.* Butterworths, 1979.

[48] Janyce M. Wiebe. Learning subjective adjectives from corpora. In *AAAI*, 2000.

Appendix A

Feature-sets

A.1 POS statistics

We use Brill's part-of-speech tagger to tag documents. This requires that the documents be in Penn Treebank format before being passed to the tagger. All html tags are removed from the document. The remaining text is converted to Penn Treebank format and passed to the tagger. The tagger returns the text with each token tagged with the appropriate POS tag. Section A.1.1 described the Penn Treebank format while section A.1.2 lists the Penn Treebank tags along with examples of each.

A.1.1 Penn Treebank Tokenisation[1]

Penn Treebank format is fairly simple.:

- there must be only one sentence per line

[1] Adapted from http://www.cis.upenn.edu/~treebank/tokenization.html

- punctuation is split from adjoining words
- double quotes (") are changed to doubled single forward and backward quotes (“ and ”) respectively for the opening and closing quotes
- verb contractions and the Anglo-Saxon genitive of nouns are split into their component morphemes, and each morpheme is tagged separately. E.g.

```
children's --> children 's
parents' --> parents '
won't --> wo n't
gonna --> gon na
I'm --> I 'm
```

There are some other subtleties for hyphens vs. dashes, ellipsis dots (...) and so on. Figure A.1 shows a function written in python to convert a string to Penn Treebank format.

A.1.2 The Penn Treebank tags[2]

Tag	Description	Examples
CC	Coordinating conjunction	and both but either et for less minus neither nor or plus so therefore times v. versus vs. whether yet

[2]Examples extracted from http://www.comp.leeds.ac.uk/amalgam/tagsets/upenn.html

CD	Cardinal number	mid-1890 nine-thirty forty-two one-tenth ten million 0.5 one forty-seven 1987 twenty ’79 zero two 78-degrees eighty-four IX ’60s
DT	Determiner	all an another any both each either every half la many much nary neither no some such that the them these this those
EX	Existential there	there
FW	Foreign word	alai je jour objets salutaris fille
IN	Preposition or subordinating conjunction	astride among upon whether out inside pro despite on by throughout below within for towards near behind atop around if like until below next into if beside
JJ	Adjective	third ill-mannered pre-war regrettable oiled calamitous first separable battery-powered participatory fourth still-to-be-named multilingual multi-disciplinary
JJR	Adjective, comparative	bleaker braver breezier briefer brighter brisker broader bumper busier calmer cheaper choosier cleaner clearer closer colder
JJS	Adjective, superlative	calmest cheapest choicest classiest cleanest clearest closest commonest corniest costliest crassest creepiest crudest cutest darkest

LS	List item marker	A A. B B. C C. D E F First G H I J K One SP-44001 SP-44002 SP-44005 SP-44007 Second Third Three
MD	Modal	can cannot could couldn't dare may might must need ought shall should shouldn't will would
NN	Noun, singular or mass	common-carrier cabbage knuckle-duster Casino afghan shed thermostat investment slide humor falloff slick wind hyena override
NNS	Noun, plural	undergraduates scotches bric-a-brac products bodyguards facets coasts divestitures storehouses designs clubs fragrances averages subjectivists apprehensions muses
NP	Proper noun, singular	Motown Venneboerger Czestochwa Ranzer Conchita Trumplane Christos Oceanside Escobar Kreisler Sawyer Cougar Yvette Ervin ODI Darryl CTCA Shannon A.K.C. Meltex Liverpool
NPS	Proper noun, plural	Americans Americas Amharas Amityvilles Amusements Anarcho-Syndicalists Andalusians Andes Andruses Angels Animals Anthony Antilles Antiques Apache Apaches
PDT	pre-determiner	all both half many quite such sure this
POS	Possessive ending	' 's

PP	Personal pronoun	hers herself him himself it itself me myself one oneself ours ourselves ownself self she thee theirs them themselves they thou thy us
PP$	Possessive pronoun	her his mine my our ours their thy your
RB	Adverb	occasionally unabatingly maddeningly adventurously professedly stirringly prominently technologically magisterially predominately swiftly fiscally
RBR	Adverb, comparative	further gloomier grander graver greater grimmer harder harsher healthier heavier higher however larger later leaner lengthier less-perfectly lesser lonelier
RBS	Adverb, superlative	best biggest bluntest earliest farthest first furthest hardest heartiest highest largest least less most nearest
RP	Particle	aboard about across along apart around aside at away back before behind by crop down ever fast for forth from go high i.e. in into just later low more off on open out over per pie raising start teeth that through under unto up up-pp upon whole with you
SYM	Symbol	< = > @ A[fj] U.S U.S.S.R *
TO	to	to

UH	Interjection	Goodbye Goody Gosh Wow Jeepers Jeesus Hubba Hey Kee-reist Oops amen huh howdy uh dammit whammo shucks heck anyways whodunnit honey golly man baby diddle hush sonuvabitch
VB	Verb, base form	ask assemble assess assign assume atone attention avoid bake balkanize bank begin behold believe bend benefit bevel beware bless boil bomb boost brace break bring broil brush
VBD	Verb, past tense	dipped pleaded swiped regummed soaked tidied convened halted registered cushioned exacted snubbed strode aimed adopted belied speculated wore appreciated
VBG	Verb, gerund or present participle	telegraphing stirring focusing angering judging stalling lactating hankerin' alleging veering capping approaching traveling besieging encrypting interrupting erasing
VBN	Verb, past participle	multihulled dilapidated aerosolized chaired languished panelized used experimented flourished imitated reunifed factored condensed sheared unsettled primed

VBP	Verb, non-3rd person singular present	predominate wrap resort sue twist spill cure lengthen brush terminate appear tend stray glisten obtain comprise detest tease attract emphasize mold postpone sever
VBZ	Verb, 3rd person singular present	bases reconstructs marks mixes displeases seals carps weaves snatches slumps stretches authorizes smolders pictures emerges stockpiles seduces fizzes uses bolsters
WDT	Wh-determiner	that what whatever which whichever
WP	Wh-pronoun	that what whatever whatsoever which who whom whosoever
WP$	Possessive wh-pronoun	whose
WRB	Wh-adverb	how however whence whenever where whereby wherever wherein whereof why

A.2 Hand Crafted Text Statistics

Feature	Description
AVE_I	Average number of sentences beginning with capital I
AVE_SEN_LEN	Average sentence length
AVE_WORD_LEN	Average word length

stopword count features	Counts of various stopwords that may be useful for genre classification. The stopwords counted are: certainly, her, highly, him, his, it, large, little, me, mine, mostly, much, my, our, probably, that, they, us, very, we, where, which, you, yours
N_I	number of sentences beginning with capital I
N_LONG_WORDS	number of long words (>5 characters)
N_WORDS	number of words in the document
punctuation counts	counts of various punctuation counts. Punctuation symbols counted are !“$%&’()*+,-.:;=?
The occurrence of various keywords in the document	The keywords used as features were absolutely, altogether, archive, article, column, completely, doesn’t, editorial, enormously, entirely, extremely, fact, feature, fully, greatly, highly, intensely, isn’t, news, opinion, perfectly, report, strongly, thoroughly, totally, utterly, very.

```
def pennTreebank(str):
    # heuristic algorithm to identify sentence boundaries
    # returns a string with a sentence on each line
    # converted to Penn Treebank style
    s = re.compile(r’([!\?\.]([\’\"])?)’).sub(r’\1\n’,str)
    #chop
    s = s[:-1]
    # attempt to get correct directional quotes
    s = re.compile(r’^\"’,re.MULTILINE).sub(r’“’,s)
    # close quotes handled at end
    s = re.compile(r’\.\.\.’,re.MULTILINE).sub(r’ ...’,s)
    s = re.compile(r’([,;:@#\\\$%&])’,re.MULTILINE).sub(r’ \1’,s)
    # Assume sentence tokenization has been done first, so split FINAL periods
    # only.
    s = re.compile(r’([^.])([.])([]\)}>\"\’]*)[ \t]*$’,re.MULTILINE).sub(r’\1 \2\3 ’,s)
    # however, we may as well split ALL question marks and exclamation points,
    # since they shouldn’t have the abbrev.-marker ambiguity problem
    s = re.compile(r’([\?!])’,re.MULTILINE).sub(r’ \1 ’,s)
    # parentheses, brackets, etc.
    s = re.compile(r’([][\(\){}<>])’,re.MULTILINE).sub(r’ \1 ’,s)
    s = re.compile(r’--’,re.MULTILINE).sub(r’ -- ’,s)
    # NOTE THAT SPLIT WORDS ARE NOT MARKED. Obviously this isn’t great, since
    # you might someday want to know how the words originally fit together --
    # but it’s too late to make a better system now, given the millions of
    # words we’ve already done "wrong".
    # First off, add a space to the beginning and end of each line, to reduce
    # necessary number of regexps.
    s = re.compile(r’$’,re.MULTILINE).sub(r’ ’,s)
    s = re.compile(r’^’,re.MULTILINE).sub(r’ ’,s)
    s = re.compile(r’"’,re.MULTILINE).sub(r’ " ’,s)
    # possessive or close-single-quote
    s = re.compile(r’([^\’])\’ ’,re.MULTILINE).sub(’\1 \’ ’,s)
    # as in it’s, I’m, we’d
    s = re.compile(r’\’([sSmMdD]) ’,re.MULTILINE).sub(’ \’\\1 ’,s)
    s = re.compile(r’\’ll’,re.MULTILINE).sub(’ \’ll ’,s)
    s = re.compile(r’\’re’,re.MULTILINE).sub(’ \’re ’,s)
    s = re.compile(r’\’ve’,re.MULTILINE).sub(’ \’ve ’,s)
    s = re.compile(r’\’LL’,re.MULTILINE).sub(’ \’LL ’,s)
    s = re.compile(r’\’RE’,re.MULTILINE).sub(’ \’RE ’,s)
    s = re.compile(r’\’VE’,re.MULTILINE).sub(’ \’VE ’,s)
    s = re.compile(r’N\’T’,re.MULTILINE).sub(’ N\’T ’,s)
    s = re.compile(r’n\’t’,re.MULTILINE).sub(’ n\’t ’,s)
    s = re.compile(r’ ([Cc])annot ’,re.MULTILINE).sub(r’ \1an not ’,s)
    s = re.compile(r’ ([Dd])\’ye ’,re.MULTILINE).sub(’ \\1\’ ye’,s)
    s = re.compile(r’ ([Gg])onna ’,re.MULTILINE).sub(r’ \1on na ’,s)
    s = re.compile(r’ ([Gg])imme ’,re.MULTILINE).sub(r’ \1im me ’,s)
    s = re.compile(r’ ([Gg])otta ’,re.MULTILINE).sub(r’ \1ot to ’,s)
    s = re.compile(r’([Mm])ore\’n’,re.MULTILINE).sub(’ \\1ore \’n ’,s)
    s = re.compile(r’ ([Ll])emme ’,re.MULTILINE).sub(r’ \1em me ’,s)
    s = re.compile(r’ \’([Tt])is’,re.MULTILINE).sub(’ \’\\1 is ’,s)
    s = re.compile(r’ \’([Tt])was ’,re.MULTILINE).sub(’ \’\\1 was ’,s)
    s = re.compile(r’ ([Ww])anna ’,re.MULTILINE).sub(r’ \1an na ’,s)
    # clean out extra spaces
    s = re.compile(r’ +’,re.MULTILINE).sub(r’ ’,s)
    s = re.compile(r’^ *’,re.MULTILINE).sub(r”,s)
    return s
```

Figure A.1: A Python function to convert a string to Penn Treebank format

Appendix B

Models built using our datasets

B.1 Introduction

This appendix gives details of the models built by C4.5 and OneR on our datasets. We can examine the models generated by C4.5 to see what features the classifier is using to make its prediction. This may give us some insight into the classification task in question and give us confidence in the model being generated. If the model gives us rules that seem intuitively related to the genre classification task, this gives us confidence in the validity of the model. The model may also give us insight into features that differentiate genre but are not obvious from inspection of the data. The classifier can give insights into the genre class under investigation.

We may also find that the classifier is using features that are a legacy of the dataset rather than indicative of the genre class.

We compare the complexity of the decision trees generated by C4.5 with those generated by OneR. OneR generated simple one level decision trees which often perform well in comparison to with the more complex decision trees generated by C4.5.

B.2 BOW

Examining the models built using the BOW feature-set we can get an idea of what words are indicative of the genre class of the document. Figure B.2 shows the OneR models generated on the subjectivity classification task in the football, politics and finance domains using the BOW feature-set. For the football domain the token 'archiv' occurring in the text indicates that the text is subjective. This is an idiosyncrasy of the dataset and does not tell us anything about the nature of subjective text. The rules generated in the other two domains are more intuitive. In the politics domain, the word 'column' indicated subjectivity while in the finance domain the word 'risk' indicates a subjective document. Figures B.10, B.11 andB.12 show the corresponding C4.5 models for the football, politics and finance domains using the BOW feature-set.

Each of these decision trees are much more complex than those generated by OneR. In each case the root of the tree is either the word 'column' or 'columnist'. This means that for the finance and football domains OneR and C4.5 choose different attributes for their root nodes. Although the word 'column' intuitively seems like a good feature for subjectivity classification, a lot of the other features seem to be unrelated to subjectivity and just a legacy of the dataset.

The performance of OneR in relation to C4.5 indicates that C4.5 is producing decision trees that are overly complex.

B.3 POS

Figure B.1 shows the OneR models generated on the subjectivity task using the POS feature-set. In the football domain, the EX tag is chosen as the most informative feature. In the other two domains, RB is chosen as the most informative feature.

Figures B.4,B.5 and B.6 show the corresponding C4.5 trees generated by the feature-set for the football, politics and finance datasets respectively. In the football domain the DT attribute is chosen as the root node of the C4.5 tree. In the other two domains, RB is chosen as the similar to the corresponding OneR models.

In the football and politics domains the trees produced are quite complex while in the finance domain the tree produced is relatively simple with a depth of 3. None of the trees produced rate the number of adjectives highly, but rather find tags such as DT, NNS, WP$, RB, NNP, WRB and WP to be most informative.

B.4 HC

Figure B.3 shows the OneR models generated for the subjectivity classification task using the hand crafted features. For the football, politics and finance domains respectively the most informative features are the presence of the word 'column', the number of words in the document and the number

```
EX:
        < 5.0E-4        -> NotOpinion
        < 0.01  -> Opinion
        >= 0.01 -> NotOpinion
(280/351 instances correct)
```

```
RB:
        < 0.0275        -> NotOpinion
        < 0.0295        -> Opinion
        < 0.0315        -> NotOpinion
        >= 0.0315       -> Opinion
(242/289 instances correct)
```

```
RB:
        < 0.0305        -> NotOpinion
        >= 0.0305       -> Opinion
(141/156 instances correct)
```

Figure B.1: OneR models generated for the football, politics and finance datasets on the subjectivity classification task (POS)

of times the word 'that' occurs.

Figures B.7,B.8 and B.9, show the corresponding C4.5 trees. For the football and finance domains, C4.5 chooses the same attributes for its root nodes as OneR. However for the politics dataset, C4.5 chooses the presence of the word 'column' as its root node.

```
archiv:
        0       -> NotOpinion
        1       -> Opinion
(272/351 instances correct)
```

```
column:
        0       -> NotOpinion
        1       -> Opinion
(240/289 instances correct)
```

```
risk:
        0       -> NotOpinion
        1       -> Opinion
(126/156 instances correct)
```

Figure B.2: OneR models generated for the football, politics and finance datasets on the subjectivity classification task (BOW)

```
column:
        0       -> NotOpinion
        1       -> Opinion
(305/351 instances correct)
```

```
N_WORDS:
        < 666.0 -> NotOpinion
        < 789.5 -> Opinion
        < 805.0 -> NotOpinion
        < 908.5 -> Opinion
        < 1326.0        -> NotOpinion
        < 1417.5        -> Opinion
        >= 1417.5       -> NotOpinion
(242/289 instances correct)
```

```
COUNT_that :
        < 7.5   -> NotOpinion
        >= 7.5  -> Opinion
(133/156 instances correct)
```

Figure B.3: OneR models generated for the football, politics and finance datasets on the subjectivity classification task (HC)

```
J48 pruned tree
------------------

DT <= 0.061: NotOpinion (87.0)
DT > 0.061
|   NNS <= 0.077
|   |   PRP <= 0.016
|   |   |   RBS <= 0: NotOpinion (17.0/1.0)
|   |   |   RBS > 0: Opinion (5.0/1.0)
|   |   PRP > 0.016
|   |   |   VBG <= 0.01
|   |   |   |   EX <= 0.003: NotOpinion (9.0)
|   |   |   |   EX > 0.003: Opinion (2.0)
|   |   |   VBG > 0.01
|   |   |   |   RB <= 0.031
|   |   |   |   |   WP$ <= 0
|   |   |   |   |   |   WRB <= 0.005
|   |   |   |   |   |   |   WDT <= 0.003: NotOpinion (11.0)
|   |   |   |   |   |   |   WDT > 0.003
|   |   |   |   |   |   |   |   CD <= 0.017: NotOpinion (2.0)
|   |   |   |   |   |   |   |   CD > 0.017: Opinion (2.0)
|   |   |   |   |   |   WRB > 0.005: Opinion (3.0)
|   |   |   |   |   WP$ > 0: Opinion (2.0)
|   |   |   |   RB > 0.031
|   |   |   |   |   DT <= 0.08
|   |   |   |   |   |   WP$ <= 0.001
|   |   |   |   |   |   |   DT <= 0.079
|   |   |   |   |   |   |   |   WRB <= 0.001: NotOpinion (2.0)
|   |   |   |   |   |   |   |   WRB > 0.001
|   |   |   |   |   |   |   |   |   RB <= 0.058
|   |   |   |   |   |   |   |   |   |   JJR <= 0.006: Opinion (20.0)
|   |   |   |   |   |   |   |   |   |   JJR > 0.006
|   |   |   |   |   |   |   |   |   |   |   CC <= 0.027: Opinion (2.0)
|   |   |   |   |   |   |   |   |   |   |   CC > 0.027: NotOpinion (2.0)
|   |   |   |   |   |   |   |   |   RB > 0.058: NotOpinion (2.0)
|   |   |   |   |   |   |   DT > 0.079: NotOpinion (3.0)
|   |   |   |   |   |   WP$ > 0.001: NotOpinion (3.0)
|   |   |   |   |   DT > 0.08: Opinion (148.0/10.0)
|   NNS > 0.077: NotOpinion (29.0)

Number of Leaves  :     19

Size of the tree :      37
```

Figure B.4: C4.5 model for football dataset on the subjectivity classification task (POS)

```
J48 pruned tree
------------------
RB <= 0.025
|   WP$ <= 0
|   |   DT <= 0.091: NotOpinion (94.0/1.0)
|   |   DT > 0.091
|   |   |   JJR <= 0.004: NotOpinion (7.0/1.0)
|   |   |   JJR > 0.004: Opinion (2.0)
|   WP$ > 0
|   |   CC <= 0.024: NotOpinion (4.0)
|   |   CC > 0.024: Opinion (3.0)
RB > 0.025
|   NNP <= 0.196
|   |   VBN <= 0.031
|   |   |   JJ <= 0.053
|   |   |   |   WDT <= 0.006
|   |   |   |   |   NN <= 0.202: Opinion (20.0/2.0)
|   |   |   |   |   NN > 0.202: NotOpinion (3.0)
|   |   |   |   WDT > 0.006: NotOpinion (3.0)
|   |   |   JJ > 0.053
|   |   |   |   SYM <= 0.004: Opinion (79.0)
|   |   |   |   SYM > 0.004
|   |   |   |   |   VBD <= 0.038: Opinion (21.0)
|   |   |   |   |   VBD > 0.038
|   |   |   |   |   |   DT <= 0.095: NotOpinion (3.0)
|   |   |   |   |   |   DT > 0.095: Opinion (2.0)
|   |   VBN > 0.031
|   |   |   VBN <= 0.032: Opinion (2.0)
|   |   |   VBN > 0.032: NotOpinion (5.0)
|   NNP > 0.196
|   |   DT <= 0.091
|   |   |   CD <= 0.012: Opinion (4.0)
|   |   |   CD > 0.012
|   |   |   |   POS <= 0.018
|   |   |   |   |   WRB <= 0.007: NotOpinion (25.0)
|   |   |   |   |   WRB > 0.007: Opinion (3.0/1.0)
|   |   |   |   POS > 0.018: Opinion (2.0)
|   |   DT > 0.091: Opinion (7.0)

Number of Leaves  :     19

Size of the tree :      37
```

Figure B.5: C4.5 model for politics dataset on the subjectivity classification task (POS)

```
J48 pruned tree
------------------

RB <= 0.028
|   WRB <= 0.004: NotOpinion (88.0)
|   WRB > 0.004
|   |   EX <= 0: NotOpinion (3.0)
|   |   EX > 0: Opinion (3.0)
RB > 0.028
|   WP <= 0
|   |   TO <= 0.02: NotOpinion (6.0)
|   |   TO > 0.02: Opinion (4.0)
|   WP > 0: Opinion (52.0/3.0)

Number of Leaves  :     6

Size of the tree :      11
```

Figure B.6: C4.5 model for finance dataset on the subjectivity classification task (POS)

```
J48 pruned tree
------------------
column = 0
|   fact = 0
|   |   news = 0
|   |   |   COUNT_my  <= 0: NotOpinion (3.0)
|   |   |   COUNT_my  > 0: Opinion (2.0)
|   |   news = 1: NotOpinion (145.0/5.0)
|   fact = 1
|   |   N_WORDS <= 626: NotOpinion (7.0)
|   |   N_WORDS > 626
|   |   |   archive = 0
|   |   |   |   extremely = 0
|   |   |   |   |   very = 0: Opinion (3.0)
|   |   |   |   |   very = 1
|   |   |   |   |   |   PUNC_" <= 3
|   |   |   |   |   |   |   AVE_SEN_LEN <= 17.311475: NotOpinion (2.0)
|   |   |   |   |   |   |   AVE_SEN_LEN > 17.311475: Opinion (6.0)
|   |   |   |   |   |   PUNC_" > 3: NotOpinion (4.0)
|   |   |   |   extremely = 1: Opinion (3.0)
|   |   |   archive = 1: Opinion (6.0)
column = 1
|   PUNC_& <= 7: Opinion (145.0/2.0)
|   PUNC_& > 7
|   |   PUNC_; <= 0
|   |   |   fully = 0: NotOpinion (18.0)
|   |   |   fully = 1: Opinion (3.0/1.0)
|   |   PUNC_; > 0: Opinion (4.0)

Number of Leaves  :     14

Size of the tree :      27
```

Figure B.7: C4.5 model for football dataset on the subjectivity classification task (HC)

```
J48 pruned tree
------------------

column = 0
|   COUNT_very  <= 2: NotOpinion (74.0/1.0)
|   COUNT_very  > 2: Opinion (4.0/1.0)
column = 1
|   PUNC_+ <= 1
|   |   PUNC_- <= 55
|   |   |   PUNC_= <= 1
|   |   |   |   PUNC_. <= 78
|   |   |   |   |   COUNT_that  <= 5
|   |   |   |   |   |   doesn't = 0
|   |   |   |   |   |   |   PUNC_. <= 41: NotOpinion (18.0)
|   |   |   |   |   |   |   PUNC_. > 41: Opinion (6.0/1.0)
|   |   |   |   |   |   doesn't = 1: Opinion (3.0)
|   |   |   |   |   COUNT_that  > 5
|   |   |   |   |   |   COUNT_him  <= 3: Opinion (129.0/8.0)
|   |   |   |   |   |   COUNT_him  > 3
|   |   |   |   |   |   |   N_I <= 0: Opinion (5.0)
|   |   |   |   |   |   |   N_I > 0: NotOpinion (3.0)
|   |   |   |   PUNC_. > 78
|   |   |   |   |   PUNC_; <= 3: NotOpinion (8.0)
|   |   |   |   |   PUNC_; > 3: Opinion (2.0)
|   |   |   PUNC_= > 1
|   |   |   |   AVE_WORD_LEN <= 5.004847: Opinion (4.0)
|   |   |   |   AVE_WORD_LEN > 5.004847: NotOpinion (12.0)
|   |   PUNC_- > 55: NotOpinion (10.0)
|   PUNC_+ > 1: NotOpinion (11.0)

Number of Leaves  :     14

Size of the tree :      27
```

Figure B.8: C4.5 model for politics dataset on the subjectivity classification task (HC)

```
J48 pruned tree
------------------

COUNT_that  <= 7
|   editorial = 0
|   |   extremely = 0
|   |   |   AVE_I <= 0.009009
|   |   |   |   COUNT_you  <= 2: NotOpinion (87.0)
|   |   |   |   COUNT_you  > 2
|   |   |   |   |   COUNT_little  <= 0: Opinion (3.0)
|   |   |   |   |   COUNT_little  > 0: NotOpinion (2.0)
|   |   |   AVE_I > 0.009009: Opinion (5.0/1.0)
|   |   extremely = 1: Opinion (2.0)
|   editorial = 1: Opinion (4.0)
COUNT_that  > 7
|   PUNC_: <= 38
|   |   article = 0
|   |   |   COUNT_you  <= 1: NotOpinion (4.0)
|   |   |   COUNT_you  > 1: Opinion (13.0/1.0)
|   |   article = 1: Opinion (31.0)
|   PUNC_: > 38: NotOpinion (5.0)

Number of Leaves  :     10

Size of the tree :      19
```

Figure B.9: C4.5 model for finance dataset on the subjectivity classification task (HC)

```
J48 pruned tree
------------------

columnist = 0
|   new = 0
|   |   1946 = 0
|   |   |   advis = 0
|   |   |   |   bignam = 0: Opinion (76.0/2.0)
|   |   |   |   bignam = 1: NotOpinion (2.0)
|   |   |   advis = 1: NotOpinion (2.0)
|   |   1946 = 1: NotOpinion (2.0)
|   new = 1
|   |   strang = 0
|   |   |   class = 0
|   |   |   |   presenc = 0
|   |   |   |   |   enforc = 0
|   |   |   |   |   |   1990 = 0
|   |   |   |   |   |   |   altern = 0
|   |   |   |   |   |   |   |   anymor = 0
|   |   |   |   |   |   |   |   |   mcgarri = 0
|   |   |   |   |   |   |   |   |   |   slow = 0
|   |   |   |   |   |   |   |   |   |   |   cost = 0: NotOpinion (171.0/3.0)
|   |   |   |   |   |   |   |   |   |   |   cost = 1: Opinion (3.0/1.0)
|   |   |   |   |   |   |   |   |   |   slow = 1: Opinion (2.0)
|   |   |   |   |   |   |   |   |   mcgarri = 1: Opinion (2.0)
|   |   |   |   |   |   |   |   anymor = 1: Opinion (2.0)
|   |   |   |   |   |   |   altern = 1: Opinion (2.0)
|   |   |   |   |   |   1990 = 1: Opinion (2.0)
|   |   |   |   |   enforc = 1: Opinion (3.0)
|   |   |   |   presenc = 1: Opinion (4.0)
|   |   |   class = 1: Opinion (4.0)
|   |   strang = 1: Opinion (7.0)
columnist = 1: Opinion (67.0)

Number of Leaves  :     16

Size of the tree :      31
```

Figure B.10: C4.5 model for football dataset on the subjectivity classification task (BOW)

```
J48 pruned tree
------------------

column = 0
|   080999 = 0
|   |   45000 = 0
|   |   |   ball = 0
|   |   |   |   clariti = 0
|   |   |   |   |   foreignpolici = 0
|   |   |   |   |   |   profound = 0: NotOpinion (112.0/2.0)
|   |   |   |   |   |   profound = 1: Opinion (2.0)
|   |   |   |   |   foreignpolici = 1: Opinion (2.0)
|   |   |   |   clariti = 1: Opinion (2.0)
|   |   |   ball = 1: Opinion (2.0)
|   |   45000 = 1: Opinion (2.0)
|   080999 = 1: Opinion (2.0)
column = 1
|   printerfriendli = 0
|   |   null = 0
|   |   |   bellweth = 0
|   |   |   |   brief = 0
|   |   |   |   |   cook = 0
|   |   |   |   |   |   skelton = 0
|   |   |   |   |   |   |   55 = 0: Opinion (131.0/2.0)
|   |   |   |   |   |   |   55 = 1: NotOpinion (3.0/1.0)
|   |   |   |   |   |   skelton = 1: NotOpinion (2.0)
|   |   |   |   |   cook = 1: NotOpinion (2.0)
|   |   |   |   brief = 1: NotOpinion (2.0)
|   |   |   bellweth = 1: NotOpinion (2.0)
|   |   null = 1: NotOpinion (5.0)
|   printerfriendli = 1: NotOpinion (18.0)

Number of Leaves  :     15

Size of the tree :      29
```

Figure B.11: C4.5 model for politics dataset on the subjectivity classification task (BOW)

```
J48 pruned tree
------------------

columnist = 0
|   your = 0
|   |   wrong = 0
|   |   |   happen = 0
|   |   |   |   pe = 0
|   |   |   |   |   2030 = 0
|   |   |   |   |   |   adam = 0
|   |   |   |   |   |   |   founder = 0: NotOpinion (101.0/1.0)
|   |   |   |   |   |   |   founder = 1: Opinion (2.0)
|   |   |   |   |   |   adam = 1: Opinion (2.0)
|   |   |   |   |   2030 = 1: Opinion (2.0)
|   |   |   |   pe = 1: Opinion (4.0)
|   |   |   happen = 1: Opinion (6.0)
|   |   wrong = 1: Opinion (10.0)
|   your = 1: Opinion (12.0)
columnist = 1: Opinion (17.0)

Number of Leaves  :     9

Size of the tree :      17
```

Figure B.12: C4.5 model for finance dataset on the subjectivity classification task (BOW)