N-gram Feature Selection for Authorship Identification

by

John Houvardas

Submitted to the University of the Aegean in partial fulfilment
of the requirements for the degree of

Master in Information Management

Supervisor	Committee Member
Dr. Stamatatos Efstathios	Prof. Vouros George
University of the Aegean	University of the Aegean

Committee Member
Dr. Kavalieratou
University of the Aegean

May 5, 2006

Contents

List of Figures

List of Tables

.

To my parents Γ*ιωργάκη* and *Κατίνα*

Acknowledgments

This is the part of a thesis where one acknowledges the contribution to his work by people close to him (people that where close, or became close during the course of work.) Their contribution can range from scientific knowledge to psychological support to just being there. Space in one page is not enough to thank every one that contributed to your work one way or another.

There is no way one can weight the help of friends who stood by him , **tfidf** [1] just does not work here. So I would first like to thank, and apologize to the friends not mentioned here, it neither means that their offer has not been appreciated nor it means that a small weight has been assigned to it.

The only contribution one could probably weight would be that of his supervisor. I have been lucky at that. I would like to thank my supervisor, Dr. Stamatatos for letting me work on this project, all his help and guidance during the course of this work as well as for his patience with me.

Working on this project was (*''like going fishing''*), to use the words of a friend mentioned below.

Thank you Stathi.

Then I would like to thank Pr. Vouros George and Pr. Likothanassis Spiros for introducing me to the field of NLP two years ago during my undergraduate adventure.

Thanks George.

Thanks Spiro.

I thank my children George and Antony for still loving me even though working on this project made me almost forget about them for the past 8 months. Popi for showing some (enough) patience with me during these last 6 years of studying. My mother *Κατινα* and my brother Θ*αναση* for encouraging me and supporting me, and last but not least all of my friends

[1] A weighting function for terms representing a document, discussed in section 3.2

who I have neglected but still stand by me encouraging me during this work. Thanks guys.

Finally I would like to thank Dr.Tsolomitis for making my life easier by introducing me to LaTeX and his continuing help with it.

John Houvardas,

April 2006,

Samos

Abstract

Automatic authorship identification offers a valuable tool for supporting crime investigation and security. It can be seen as a multi-class, single-label text categorization task.

Automatic authorship identification depends on selecting stylistic features that would capture an authors writing style independent of the content or genre of text. Character n-grams have been used successfully to represent text for stylistic purposes in literature. They seem to be able to capture nuances in lexical, syntactical, and structural level. To date character n-grams of fixed length have been used for authorship identification.

In this thesis:

- we propose the use of variable-length n-grams to represent the stylistic information of the documents to be classified.
- we introduce a new approach for selecting variable length n-grams inspired by previous work for selecting variable-length word sequences.
- we explore the significance of digits as stylistic features for distinguishing between authors and show that an increase in performance can be achieved using simple text pre-processing.

Using a subset of the new Reuters corpus, consisting of texts on the same topic by 50 different authors, we show that the proposed feature selection method is at least as effective as information gain for selecting the most significant n-grams although the feature sets produced by the two methods have few common members.

Introduction

Since early work on 19th century, authorship analysis has been viewed as a tool for answering literary questions on works of disputed or unknown authorship. In recent years, researchers have paid increasing attention to authorship analysis in the framework of practical applications, such as verifying the authorship of emails and electronic messages, plagiarism detection, and forensic cases. Authorship identification is the task of predicting the most likely author of a text given a predefined set of candidate authors and a number of text samples per author of undisputed authorship. From a machine learning point of view, this task can be seen as a single-label multi-class text categorization problem where the candidate authors play the role of the classes.

Problem Definition

One major subtask of the authorship identification problem is the extraction of the most appropriate features for representing the style of an author. Several measures have been proposed, including attempts to quantify vocabulary richness, function word frequencies and part-of-speech frequencies. The vast majority of proposed approaches are based on the fact that a text is a sequence of words. Treating text as a sequence of words is an approach prone to errors and language dependent.

A promising text representation technique for stylistic purposes, that is not affected by this fact is the use of character n-grams.

Character n-grams are able to capture complicated stylistic information on the lexical, syntactic, or structural level. The problem with this representation is the dimensionality of the feature space produced. Due to this fact, n-grams of fixed length have been used so far (e.g. 3-grams).

Our Approach

To solve the problem we propose the following solutions.

Variable Length n-grams We introduce the use of variable length n-grams for the task of Authorship attribution. Works presented in literature have proven that the selection of the optimal length of n-grams to be used is language dependent. The best results to date have been achieved using fixed length n-grams of size 3, 4 **or** 5.

To get the most out of the n-gram approach and keep the feature set to a size that can be handled by machine learning algorithms we use variable length n-grams of size 3, 4 **and** 5 and introduce a new feature selection method that will reduce the size of the feature set to an acceptable size.

Feature Selection: We introduce a new Feature Selection method for variable-length n-grams. The original idea is based on previous work for extracting multiword terms (word n-grams of variable length) from texts in the framework of information retrieval applications [14, 15].

Text Preprocessing: We examine a simple pre-processing procedure for removing redundancy in digits found in texts. It is shown that this procedure improves the performance of the proposed approach.

Thesis Layout

The thesis is organized as follows

Chapter 1: An introduction to Text Classification and its subtasks, Content based and Style based classification.

Chapter 2: An introduction to Classification methods, Machine learning and a brief review of algorithms used for the task. An extensive pre-

sentation of SVMs,the algorithm used to test the effectiveness of our method.

Chapter 3: We examine the methods used to date to represent style, as well as the problems associated with them, the advantages and disadvantages of each method. We discuss variable length n-grams and their ability to represent stylistic features. We also examine the effects of simple text preprocessing prior to feature selection.

Chapter 4: We discuss the methods used to select the features that will be used to build the classifiers. We discuss Information Gain (the feature selection method that has been found to be the most effective by Yang et al. [5]) and the reasons why it is not the most appropriate method for stylistic features selection. We present our method and analyze the reasons it is better suited for the task at hand.

Chapter 5: We present experiments in Authorship Identification using SVMs to test the efficiency of Information Gain and the proposed method against it. We present the Corpus used. We discuss the effect of text preprocessing and present the results of experiments conducted with preprocessed text.

Chapter 6: Conclusions drawn out of our experiments are presented as well as our plans for future work.

Conclusions

For any feature selection method to be considered as a promising new method it would have to produce results at least as good or better than Information Gain. In this work we have introduced a feature selection method for variable length n-grams for Author Identification that attained at least as good results as Info Gain for large numbers of features and much better when small numbers of features where used (< 4000). Our method based

on work by Silva et al. [3] also managed to get those results using different features than the ones Info Gain did. An examination of the features selected showed that even when the two methods attained comparable results they did so using sets of features with very few common members. Using this method we where able to effectively select variable length n-grams for the task.

It was also proven that simple text preprocessing and in particular replacing digits with a special character, did help our feature selection method pick *better* features, improving categorization results for almost all of our experiments.

Chapter 1

Text Classification

Document Classification (or Categorization) is described in [11] as the task of determining an assignment of a value from {0.1} to each entry of the *decision matrix*

	d_1	...	...	d_j	...	...	d_n
c_1	a_{11}	...	...	a_{1j}	...	...	a_{1n}
...	...	...	...	...	...	...	...
c_i	a_{i1}	...	...	a_{ij}	...	...	a_{in}
...	...	...	...	...	...	...	...
c_m	a_{m1}	...	...	a_{mj}	...	...	a_{mn}

where $C = \{c_1, \ldots, c_m\}$ is a set of pre-defined *categories* and $D = \{d_1, \ldots, d_n\}$ is a set of documents to be classified. A value of 1 for a_{ij} is interpreted as a decision to file d_j under c_i while a value of 0 is interpreted as a decision not to file d_j under c_i.

Fundamental to the understanding of this task are two observations:

- the categories are just symbolic labels. No additional knowledge of their "meaning" is available to help in the process of building the classifier; in particular, this means that the "text" constituting the label (e.g Sports in a news categorization task) can not be used;

- the attribution of documents to categories should, in general, be attributed on the basis of the *content* of the documents, and not on the basis of *metadata* (e.g publication date, document type, etc.) that may be available from an external source.

A Category label can be assigned to a document according to its content or style

1.1 Content

Content based classification is concerned in assigning a label to a document based on its content. It has been used in many applications such as:

- Automatic Indexing for Boolean Information Retrieval Systems
- Document Organization
- Text Filtering
- Word Sense Disambiguation
- Hierarchical Categorization of Web Pages

1.2 Style

1.2.1 Genre

> Genre is necessarily a heterogeneous classificatory principle, which is based among other things on the way the text was created, the way it is distributed, the register of language it uses, and the kind of audience it is addressed to. For all its complexity, this attribute can be extremely important for many of the core problems that computational linguists are concerned with. (Kessler [28]).

Genre Classification enables people to search for documents according to their interests. A library search engine could not only provide information about the topics of the books or articles available but for their genre also.

A scientist seeking information about the works of Plato would have completely different requirements from a casual reader who wants to learn about Greek philosophers. Genre classification can play an important role in spam identification.

1.2.2 Author

Humans are creatures of habit and have certain personal traits which tend to persist. All humans have unique (or near unique) patterns of behavior, biometric attributes, and so on. We therefore conjecture that certain characteristics pertaining to language, composition and writing, such as particular syntactic and structural layout traits, patterns of vocabulary usage, unusual language usage (eg. converting the letter ”f” to ”ph”, or the excessive use of digits and/or upper-case letters), stylistic and sub-stylistic features will remain relatively constant.

The identification and learning of these characteristics are the principal challenges in authorship categorization.

Authorship identification has been used in a small but diverse number of application areas.

The first time Machine Learning was used in authorship categorization was by Mosteller and Wallace [10] who managed to append authorship to the 12 Federalist papers in dispute.

The Federalist Papers were written in 1787-1788 by Alexander Hamilton, John Jay and James Madison to persuade New York to ratify the United States Constitution. They were published anonymously, and as a result, although some of the 85 essays were clearly attributable to one author or another, the authorship of 12 were in dispute between Hamilton and Madi-

son.

Program code authorship has been researched by scientists in the context of software theft, software plagiarism, intrusion detection and malicious code authorship identification [35].

Author Identification

Authorship identification is the task of determining the author of a piece of text. Since early work on 19th century, authorship analysis has been viewed as a tool for answering literary questions on works of disputed or unknown authorship.

From a machine learning point of view, this task can be seen as a single-label multi-class text categorization problem where the candidate authors play the role of the classes.

Author Verification

Sometimes we need to determine whether a given author (for whom we have a corpus of writing samples) is also the author of a given anonymous text. The set of alternate candidates is not limited to a given finite closed set. In this case while we can have an unlimited set of negative examples, we can never be sure that these negative examples represent the space of all alternative authors.

Recently, increasing attention is payed to authorship verification in the framework of practical applications, such as verifying the authorship of emails and electronic messages, plagiarism detection, and forensic cases.

The forensic analysis of text attempts to match text to authors for the purpose of criminal investigation.

Chapter 2

Classification Methods

The increase of the availability of documents in digital form in the last 10 years has raised a big interest in Automatic Classification methods. Knowledge Engineering and Machine Learning Approaches have been introduced to handle the problem.

2.1 Knowledge Engineering

The first approach to Automatic Classification of text was using Knowledge-engineering. A *knowledge* engineer had to build a set of rules with the aid of a domain expert. This set of rules would be of type

```
if(DNF Boolean formula) then
 file in category C
endif
```

where DNF stands for *Disjunctive Normal Form*.

The drawback of this "manual" approach to the construction of classifiers is the existence of a *knowledge acquisition bottleneck*, similar to the one that happens in expert systems. The rules that where defined would have to be reconstructed in case the set of categories are updated[1] or the

[1]Some categories eliminated, renamed, or new ones added

classifier is used in a different domain.

An example of this approach is the CONSTRUE system built by the Carnegie Group for use at the Reuters news agency. Results outperforming all automatic classifiers using machine learning techniques have been reported on a subset of Reuters-21578 test collection. However no other classifier has been tested on the same data set and it is not clear how this data set was selected from the Reuters-21578 collection (i.e whether it was a random or a favorable subset of the whole collection).

2.2 Machine Learning

Since the early 1990's a new approach has gained popularity in the scientific community and has become the dominant one. The Machine learning approach.

Informal Definition: A machine learning algorithm is any computer program that improves its performance at some task through experience and/or data.

Formal definition: A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E. (Stamatatos [19])

The machine learning approach relies on the availability of an initial corpus of documents preclassified by a domain expert.

The corpus is split in two sets the *training set* used to build the classifier and the *test set* used to test the efficiency of the classifier built.

Efficiency testing is carried out by feeding the classifier each one of the documents of the test set and comparing its decision with the domain experts decision.

The documents belonging to the *test set* should not participate in any way in the construction of the classifiers. The latter is not enforced in case there is a very limited corpus. I that case techniques have been devised (cross validation etc) to avoid biasing the classifiers.

The classification problem is an activity of supervised learning, since the learning process is supervised by the knowledge of the categories and of the training instances that belong to them.

To use machine learning algorithms an indexing procedure has to take place that will map training and test documents into a compact representation that can be handled by the algorithm. The choice of a representation for text (see chapter 3) depends on what are regarded as the meaningful units of text for the classification task at hand.

Terms have to be weighted by some statistical function according to their contribution towards the detection of a category and feature selection applied to reduce the dimensionality of the feature space (see chapter 4) to a size that can be handled by the ML algorithm.

A number of learning approaches have been applied including Bayesian probabilistic approaches, decision trees, inductive rule learning and Support Vector Machines.

2.2.1 k-Nearest Neighbor

A well known statistical approach that has been intensively studied and used [5],[18] in pattern recognition and text categorization [22].

It belongs to the group of classifiers called *lazy classifiers* because, they

> ”they defer the decision on how to generalize beyond the training data until each new query instance is encountered” [23]

These are example based classifiers and do not build an explicit, declarative representation of the category, but rely on category labels attached to

the training documents similar to the test document. To decide whether a document d_j belongs in category c_i kNN checks whether the k training documents most similar to d_j are also in c_i. If a large enough portion of them do belong in c_i a positive decision is taken or a otherwise a negative decision is taken.

2.2.2 Neural Networks

> A *neural network* text classifier is a network of units, where the input units represent terms, the output unit(s) represent the category or categories of interest, and the weights on the edges connecting units represent dependance relations [12].

To classify a document weights are assigned to the terms (usually words) representing it and those weights are loaded to the input units. These weights are then propagated through the network and the range the output is in determines the category of the document in question.

Training of the network usually takes place with the back-propagation method. Documents from the training set are fed to the network and the weights of their terms are propagated through the network, if the result does not fall within the predefined range, the mistake is back-propagated through the network and the weights of the network connections are readjusted again and again until the network *learns* (or the error is minimized). Neural Networks have been successfully used in TC [24]. A problem faced when using NNs is that time taken to train the network increases with the number of features used and it becomes unacceptable for big numbers of features.

2.2.3 Decision Tree

Decision Tree is a well-known machine learning approach to automatic induction of classification trees based on training data. Applied to Text Categorization, Decision Tree algorithms are used to select informative words

based on an information gain criterion, and predict categories of each document according to the occurrence of word combinations in the document.

2.2.4 Naive Bayes

Naive Bayes probabilistic classifiers are also commonly-used in text categorization. The basic idea is to use the joint probabilities of words and categories to estimate the probabilities of categories given a document. The naive part of such a model is the assumption of word independence. The simplicity of this assumption makes the computation of the Naive Bayes classifier far more efficient than the exponential complexity of non-naive Bayes approaches because it does not use word combinations as predictors.

There are several variants of naive Bayes classifiers, including the binary independence model, the multinomial model, the Poison model, and the negative binary independence model. It has been shown that for text classification applications, the multinomial model is most often the best choice [37], [38].

2.2.5 Support Vector Machines

Support Vector Machines where first introduced in Text Classification by Joachims [6] and subsequently used by many researchers of the field. It is the dominant method in use to date.

Lodhi et al. [34] used SVMs for text categorization with good results. De Vel et al. used SVMs for E-mail Author Identification Forensics [15].

Support vector machines are based on an algorithm that finds a special kind of linear model: the *maximum margin hyperplane*. To visualize a maximum margin hyperplane, we can think of a two-class dataset whose classes are linearly separable, that is, there is a hyperplane in instance space that classifies all training instances correctly. The maximum mar-

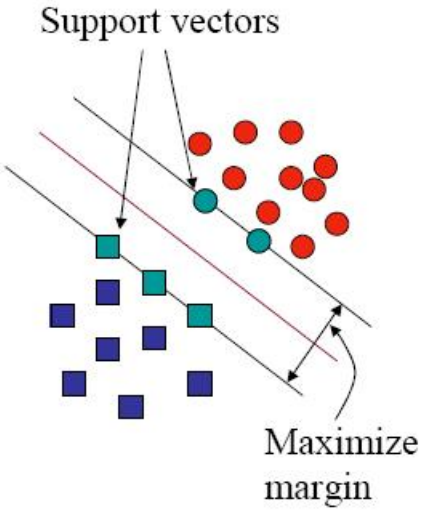

Figure 2.1: The decision line with the maximum margin. The data points crossed by the two parallel lines are the support vectors

gin hyperplane is the one that gives the greatest separation between the classes, meaning that it comes no closer to either than it has to. An example is shown in figure 2.1 in which the classes are represented by squares and circles respectively.

Technically , the convex hull of a set of points is the tightest enclosing convex polygon, its outline emerges when you connect every point of the set to every other point. Because we have supposed that the two classes are linearly separable, their convex hulls cannot overlap. Among all hyperplanes that separate the classes, the maximum margin hyperplane is the one that is as far away as possible from both convex hulls. This hyperplane is the perpendicular bisector of the shortest line connecting the hulls.

The instances that are closest to the maximum margin hyperplane are called *support vectors*. There is always at least one support vector for each class and often there are more. The important thing is that the set of support vectors uniquely defines the maximum margin hyperplane for the

learning problem. Given the support vectors for the two classes, we can easily construct the maximum margin hyperplane. All other training instances are irrelevant, they can be deleted without changing the position and orientation of the hyperplane.

Finding the support vectors for the instance set belongs to a standard class of problems known as *constrained quadratic optimization*.

SVMs are universal learners. In their basic form they learn linear threshold function. However by just a simple "plug-in" of an appropriate Kernels they can be used to learn polynomial classifiers, radial basic function (RBF) networks, and three layer sigmoid neural nets.

A very important property of SVMs is that their ability to learn can be independent of the feature space.

General Properties of Text

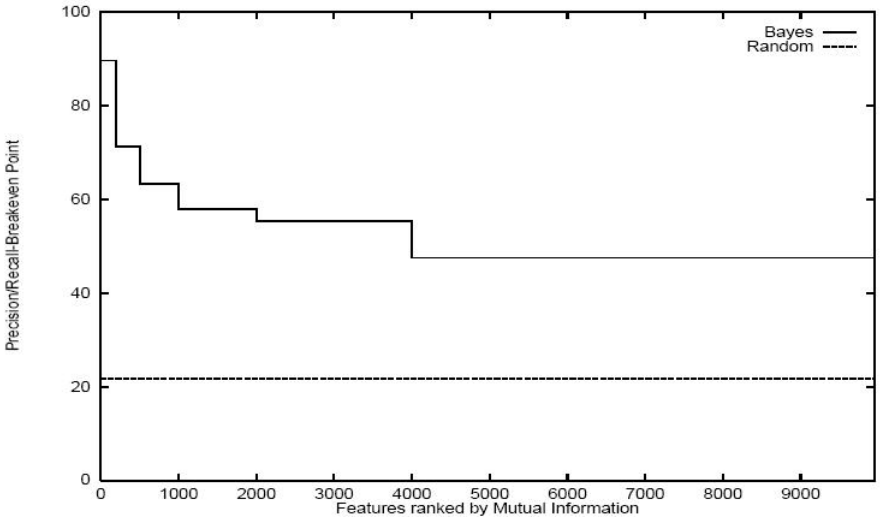

Figure 2.2: Learning without using the "best" features

To investigate if SVMs are the appropriate method for learning text classifiers we look at the properties of text.

High dimensional input space: One of the problems encountered with learning to classify text is the dimensionality of the feature space. SVMs use overfitting protection and have the potential to handle these large feature spaces.

Few irrelevant features: Joachims in [6] experimented on the Reuters "acq" category using a subset of the features that where not ranked highly according their information gain. A naive Bayes classifier was trained using only those features ranked 1-200, 201-500, 501-1000, 1001-2000, 2001-4000, and 4001-9962. The results in figure 2.2 show that even features ranked lowest still contain considerable information and are somewhat relevant. A classifier using only those "worst features" had a performance better that random. Since it seems unlikely that all those features are redundant, this leads to the conjecture that a good classifier should combine many features.

Document vectors are sparse: For each document, the corresponding document vector contains only few entries which are not zero. Theoretical and empirical evidence has been given [20] for the mistake bound model that "additive" algorithms, which have a similar inductive bias like SVMs, are well suited for problems with dense concepts and sparse instances.

Most text categorization problems are linearly separable: Many of the Reuters categories are linearly separable. The idea of SVMs is to find such linear (or polynomial, RBF, etc.) separators.

For the above reasons SVMs should perform well for text categorization.

Chapter 3

Text Representation

Anytime a linguist leaves the group the recognition rate goes up.

Fred Jelinek (then of the IBM speech group)(1988)[1]

To use Machine Learning Algorithms we have to find the most appropriate form or representation for the texts we want to classify and for the task we want to undertake. A vector of n index terms (usually weighted) is used to represent each text or/and category.

As one would expect a different kind of representation is needed depending on the categorization task.

Topic based classification would need features representing the content of the texts in question.

Genre and Authorship classification would require features representing the style of the text.

3.1 Representing Style

Selecting the appropriate features to represent Genre of Author style is one of the major problems that a stylometrist would have to solve. As Rudman

[1] In an address to the first Workshop on the Evaluation of Natural Language Processing Systems, December 7,1988.

[17] points out, 1000 style markers have already been identified. Rudman also points out that all style markers have to be found and style has to be mapped the same way biologists map gene. Many of these markers have already been used in the literature.

In this section we examine the ones that are most frequently used (Stamatatos et al.[1]).

Author

The statistical analysis of style, *stylometry*, is based on the assumption that every author's style has certain features inaccessible to conscious manipulation. These features provide the most reliable basis for the identification of an author.

However, the style of an author may very well vary as a result of differences in topics or genre, or the personal development of the author over time. It may also be influenced by the explicit imitation of literary style. Ideally stylometry should identify features which are invariant to these effects, but are expressive enough to discriminate an author from other writers.

Stamatatos et al. in [2] used existing NLP tools employing various stylistic features for authorship identification of authors of Greek news stories.

3.1.1 Token-Level Measures

Viewing the text as a set of tokens grouped in sentences is the easiest approach to use. Typical measures of this category are word count, sentence count, character per word count, punctuation marks count. These features have been widely used in both Genre and Author identification.

3.1.2 Syntactic Annotation

Measures related to syntactic annotation are commonly used in text genre detection. Some of these measures are passive count, nominalization count,

and counts of frequency of various syntactic categories. Their calculation requires tagged of parsed text. Current NLP tools are not able to provide accurate calculation results for many of the proposed style markers.

3.1.3 Vocabulary Richness

Vocabulary richness is typically measured as the ratio V/N where V is the size of the vocabulary of the sample text, and N is the number of tokens of the sample test. Hapax legomena (words appearing once in a text) and dislegomena (words appearing two times in a text) have also been used as vocabulary richness measures.

Vocabulary richness has been use in conjunction with other stylistic features to achieve better results. Recent studies have proved that most of the vocabulary richness functions are text length dependent and unstable for texts shorter than 1,000 words.

3.1.4 Common Word Frequencies

Common word frequencies have been applied to text genre detection as well as authorship identification successfully. I is an approach that needs fine tuning depending on the corpus and the language used. Words that best distinguish a given group of authors can not be applied to a different group of authors with the same success.

3.2 Representing Content

The usual approach to representing content is the "*bag of words*" approach. Function words are usually removed from the feature set and sometimes stemming takes place before the word features are weighted and added to the feature vector.

In general, for determining the weight w_{jk} of term d_j in document d_j

any IR-style indexing technique that represents a document as a vector of weighted terms have been used. Most of the times, the standard $tfidf$ weight function is used, defined as

$$tfidf(t_k, d_j) = \#(t_k, d_j).\log \frac{|T_r|}{\#(t_k)} \tag{3.1}$$

3.3 Words

Words are the fundamental building blocks of text. Sequences of words ordered and connected according the language's syntactic rules form the sentences of a text. Words carry information about the content and style of a document. They have been used to represent text for topic classification as well as author and genre classification.

Words can be useful or redundant features, depending on the task at hand. Functions or stop words offer no information for Topic classification and are usually removed. Words are stemmed to shrink the feature space used for topic classification.

On the contrary function words and word endings do carry a lot of stylistic information that have been used for Authorship as well as genre classification.

To use words as features, text has to be tokenized.

3.3.1 Tokenization Problems

There are a number of features of text in human languages that make it difficult to process automatically making the task of extracting the features needed for classification purposes a difficult task.

Periods

Words are not always separated by white space. Often punctuation marks appear attached to words, such as commas, semicolons, and periods. The

first thought coming to mind is just remove punctuation marks and we do not to have about them any more, or use them to separate sentences or paragraphs.

Things are no that simple.

While most periods indeed signify the end of a sentence, others mark an abbreviation. These abbreviation periods should remain as a part of the word, and in some cases that is important. A period distinguishes *Wash.*, an abbreviation for the state of Washington, from the verb *Wash*. When an abbreviation like *etc.* appears at the end of a sentence, the only one period occurs, but it serves both functions of a period, simultaneously.

Single apostrophes

One of the problems a tokenizer is faced with is treating contractions as *I'll* or *isn't*. Some tokenizers treat them as a single word but others treat them as two separate words and expand them. If contractions are not expanded one will end up with funny words in their feature set, words as *'s* and *n't*. Expanding contractions would alter stylistic features.

Phrases as the *the dog's* and *the child's*, when not abbreviations for *the child is* or *the dog has* are commonly seen as containing *dog's* as the genitive or possessive case of *dog*.

We could go on and on about different cases in English and things would be chaotic if we start examining other languages too.

Hyphenation

Perhaps the most difficult problem arises when dealing with hyphens in the input. Hyphens are used to break words to improve justification of text, some times to join two words that should actually be treated as one *(e-mail, co-operate)*, to help indicate the correct grouping of words *The text-based medium, the 90-cent-an-hour raise*, in which case we would want things

separated by hyphens treated as separate words.

Some times there is great inconsistency in the way hyphens are used. In the Dow Jones newswire, one can find all of, *database, data-base* and *data base* (the first and the third are commonest, with the former appearing to dominate in software contexts, and the third in discussions of company assets, but without there being any clear semantic distinction in usage). These different usages of hyphenation does include important stylistic information that can be lost.

Word segmentation in different Languages

Ancient Greek was written by Ancient Greeks without separating spaces. Many modern languages, such as East-Asian Languages/scripts do not put spaces in between words. A modern tokenizer would be of no help with these languages.

German compound nouns are written as a single word, for example *Lebensversicherungsgesellschaftsangestellter* "life insurance company employee". Joining compounds sometimes happens in English too especially when they are common and have a specialized meaning. One can find in a text the word *database* or *data base*, *harddisk* or *hard disk*.

3.4 N-grams

3.4.1 Word n-grams

Word n-grams are sequences of n adjacent words. Word n-grams have been excessively used in many NLP applications. Guessing the next word (or **word prediction**) is an essential subtask of speech recognition [8], hand-writing recognition, augmentative communication for the disabled, and spelling error detection. In such tasks, word identification is difficult because the input is very noisy and ambiguous. Thus looking at previous

Table 3.1: Growth in numbers of parameters for n-gram models

Model	Parameters
2-gram model	$20,000 \times 19,999 = 400$ million
3-gram model	$20,000^2 \times 19,999 = 8$ trillion
4-gram model	$20,000^3 \times 19,999 = 1.6 \times 10^{17}$

words can give us an important cue about what the next ones are going to be.

This ability to predict the next word is important for **augmentative communication systems**. These are computer systems that help the disabled in communication. For example people who are unable to use speech or sign-language to communicate, like the physicist Steven Hawking, use systems that speak for them, letting them choose words with simple hand movements, either by spelling them out, or by selecting words from a menu of possible words. But spelling is very slow, and a menu of words obviously can't have all possible words on the screen. Thus it is important to be able to know which words the speaker is likely to want to use next, so as to put those on the menu.

Using n-gram models to represent text does produce a lot of parameters to be considered. For instance, if we conservatively assume that a corpus contains a vocabulary of 20,000 words, the we get estimates for numbers of parameters shown in table 3.1.

To extract word n-grams text has to be tokenized thus making the method language dependent and complicated.

3.4.2 Character n-grams

A character n-gram is a sequence of n adjacent characters.

Here is a sequence of seven japanese characters:

社長兼業務部長

Since Japanese doesn't have spaces between words, one is faced with the initial task of deciding what the component words are. In particular, this character sequence corresponds to at least two possible word sequences, "president, both, business, general manager" (="a president as well as a general manager of business") and "president, subsidiary-business, Tsutomu (a name), general-manager" (=?). It requires a fair bit of linguistic information to choose the correct alternative.

Countless examples as the above can be presented proving that it is impossible to use existing parsers and tokenizers for some languages. Tokenization is a problem further described in section 3.3.1.

Character level n-grams need no taggers, parsers, tokenizers or any language dependent and non-trivial NLP tools. The extraction of n-grams is a language independent task, making the approach feasible for almost all categorization problems.

Character n-grams are able to capture complicated stylistic information on the lexical, syntactic, or structural level. For example, the most frequent character 3-grams of an English corpus indicate lexical (|the|, |␣to|, |tha|), word-class (|ing|, |ed␣|), or punctuation usage (|.␣T|, |␣"T|) information. Character n-grams have been proved to be quite effective for author identification problems. Kešelj et al. [4] tested this approach in various test collections of English, Greek, and Chinese text, improving previously reported results. Moreover, a variation of their method achieved the best results in the ad-hoc authorship attribution contest [21], a competition based on a collection of 13 text corpora in various languages (English, French, Latin, Dutch, and Serbian-Slavonic). The performance of the character n-gram approach was remarkable especially in cases with multiple candidate authors (> 5).

The problem with character n-grams representation is the size of the feature space produced.

The training set we are using for our experiment consisting of $2,500$ small texts produced a total of $12,597,804$ different n-grams of sizes 2 to 11 (table 3.2). This is an almost impossible size of feature space to be handled by any one of the existing machine learning algorithms.

Decisions have to be made about the size of n-grams chosen to represent the problem as well as the number of n-grams to be used. A feature selection method has to be applied to select the most important features that will represent a class.

Lodhi et al. [34] used character n-grams of fixed length to for topic categorization using SVMs with good results.

Peng et al. [39] used fixed character n-grams and word n-grams for Authorship Attribution as well as topic categorization.

The usual approach to date for the task of Authorship attribution would be choosing among one of 3-gram, 4-gram or 5-gram representations.

In this thesis we are introducing the use of variable length n-grams for the task of Authorship Identification. We are using 3-grams,4-grams and 5-grams together because they are the sizes that have been used in literature with the best classification results to date.

3.5 Preprocessing text

Text pre processing is the act of altering text prior to extracting the features used to represent it. Text elements not useful for the task at hand are usually removed or replaced before the actual processing of the text takes place.

Such elements can be XML or HTML tags. These tags are sometimes irrelevant to the text content or style, or contain meta-data that can not be used for classification tasks. In other cases these same elements can

Table 3.2: Corpus n-grams

n	number of n-grams
2	3,000
3	26,767
4	115,488
5	315,553
6	655,697
7	1,127,202
8	1,691,381
9	2,298,435
10	2,899,521
11	3,464,760
Total	12,597,804

provide information (hyperlinks etc) used in the feature set.

Depending to the representation method to be applied such features are either removed or used.

3.5.1 Digits

Digits are a good source of information about the topic (financial reports) and or genre (press reportage, press editorial, official documents, etc.) of a piece of text. The information represented by digits may correspond to financial information, dates, values, telephone numbers etc.

In the case of authorship classification the information contained in digits is the actual use of them and not the different combinations of digits used. Hence, replaced all digits with a special character like ('@') will help avoid many of redundant n-grams keeping the information of the presence of digits. For example, all |1999|, |2000|, |2001|, and |2002| 4-grams

would be replaced by |@@@@|. Frequent use of this transformed 4-gram could be due to frequent reference to dates. In this work we study the effect of pre-processing texts for removing redundant digit characters. The use of digits is rather associated with text-genre and topic than authorship.

Digits present in text could confuse classifiers if the texts used for authorship attribution came from different thematic categories.

Digits do offer stylistic information about the author of a text, as some authors like to use numbers in their texts and others do not.

Chapter 4

Feature Selection

A major drawback of using character ngrams to represent text is the high dimensionality of the feature space produced (see table 3.2). The numbers of ngrams produced are in the neighborhood of hundreds of thousands or even millions. Not many categorizing algorithms can cope well with this size of feature space.

Automatic feature selection algorithms are assigned with the task of removing redundant features. Features that do not offer information about the category have to be removed and sometimes new features have to be produced. Corpus statistics have to be computed and the importance of presence or absence of each feature determined. A good Feature Selection algorithm should keep features that offer information about the category and eliminate redundant.

Information-theoretic functions have been put to use for the task of Feature Selection. Yang et al. [5] extensively tested most of the frequently used algorithms.

Document frequency thresholding (DF), Information Gain (IG), Mutual Information (MI), χ^2 statistic (CHI) and Term Strength (TS) where used to extract features of the Reuters-22173 and OHSUMED collections. To test the effectiveness of the methods two classifiers where used, a kNN classifier,

and a regression method named Linear Least Squares Fit Mapping.

Information Gain and CHI where found to be the most effective for aggressive feature removal.

4.1 Information Gain

Information gain measures the number of bits of information gained about a category by knowing the presence or absence of a term in a document. It is calculated as:

$$IG(t_k, c_i) = \sum_{c \in \{c_i, \bar{c}_i\}} \sum_{t \in \{t_k, \bar{t}_k\}} P(t, c). \log \frac{P(t, c)}{P(c), P(t)}$$

Information Gain examines the bits of information gained for the category for each feature alone; to decide about a feature neither it takes into account nor it examines the other features present in the set. This fact does not pose a problem when words or fixed length n-grams are used as features.

When variable length n-grams are used for classification a new problem arises. An n-gram of size n that reduces the uncertainty of a category can be contained in another (n+k)-gram, or it can contain an (n-k)-gram that has also been found to be important by Info Gain. There can be in up to (k-n+1) n-grams of size n that come from the same k-gram. Most of these important n-grams will be included in the set of features selected by Info Gain, leaving out features less important, that otherwise would be selected (if there was room left for them).

When 5000 n-grams (3-grams, 4-grams and 5-grams) are selected using Info Gain, 3-gram ”and” appears 27 times, most of them as part of the same higher order n-gram (found in 4-grams and 5-grams) i.e. ”␣and”, ”␣and␣”, ”␣hand”, ”hand␣”, ”and␣” etc. IG does not avoid selecting the same n-gram over and over again.

4.2 Document Frequency Thresholding

Document frequency is the number of documents in which a term occurs. The method is based on a simple idea and has produced good results for selecting features for text classification [5].

The frequency of each unique feature in the training set is computed and the terms that appear less than some predetermined threshold are removed. The method is based on the assumption that a term not appearing often in the training set is not important for the category and thus does not influence the classifiers performance. It is easy to use in very large corpora with a computational complexity approximately linear in the number of training documents.

However it is not frequently used because it has been proven that terms appearing seldom in a corpus are also informative and should not be removed aggressively.

4.3 Mutual Information

Mutual information is the reduction of uncertainty of one random variable due to the knowing about another, or in other words, the amount of information one random variable contains about another. It has been used extensively in NLP in a modified version that measures the mutual information of instances of those variables and is discussed in section 4.5.4.

4.4 χ^2 statistic (CHI)

The χ^2 statistic measures the lack of independence between document t and category c.

Using the two-way contingency table of a term t and a category c, where A is the number of times t and c co-occur, B is the number of times t occurs

without c, C is the number of times c occurs without t, D is the number of times neither c nor t occurs, and N is the total number of documents, the term-goodness measure is defined as:

$$\chi^2(t,c) = \frac{N \times (A \cdot D - C \cdot B)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)} \tag{4.1}$$

The χ^2 statistic has a value of 0 if t and c are independent. The computation of χ^2 scores has a quadratic computational complexity.

4.5 Our Approach

Having decided to use variable length n-grams we needed to find a way to select n-grams avoiding to select the same one again and again (found in higher of lower order n-grams). None of the feature selection methods appearing in the literature was appropriate for the task. We needed an algorithm that would decide if a feature gets selected by comparing its contribution to the classification task with other features.

Silva et al. have used an algorithm that computes local maxima to extract Multi Word Units from text. The algorithm compares the *glue* (section 4.5.2) of word n-grams with the glue of n-grams that contain them and are contained in them and selects the ones that have the maximum glue holding them together. The calculation of glue holding n-grams has been used in word clustering and for the extraction of collocations and word bigrams.

Wang et al. [36] used LocalMaxs to automatically extract key information from sensitive text documents for intelligence analysis.

4.5.1 Multi Word Units

Silva et al in their paper *"A Local Maxima method and a Fair Dispersion Normalization for extracting multi - word units from corpora"*[3] introduce a new algorithm for extracting *Multi Character Units* from text.

Multi Character Units are: Compound nouns, Compound verbs, Compound prepositions, Compound conjunctions, Frozen forms etc. These *word n-grams* appear to have a high degree of *(glue)* bonding them together.

In the same paper they introduce:

- A new way to measure the *glue* that holds the words of an ngram.
- A new algorithm for normalization, *fair dispersion point normalization*, that increases the precision and recall of the MWUs that are produced by those algorithms.

In this work we alter and adjust those algorithms to produce **Multi character units**.

4.5.2 Multi Character Units

Multi Character Units are a subset of the character n-grams found in a text. An n-gram is considered to be an MCU if there is a certain amount of *glue* holding its characters together. Let us give an example of that. Look at a frequent MCU **'the'**.

In our corpus the most frequent 3-gram is "th␣" appearing 91042 times followed by "he␣", appearing 76764 times and "the", appearing 76470 times.

The observed frequencies make it obvious that *'the'* should be considered as an MCU, since when **'th'** appears it is highly possible that **"e"** will follow, or when **'he'** appears it is highly possible that **'t'** appeared attached to its left side.

It is possible to find other characters between the characters of an MCU as in the string "...**th**is, w**e**.." or "**t**oday **he**.." meaning that there could be some *dispersion* between the characters, but still there is a lot of *glue* holding them together. Enough to think of them as an MCU .

Pseudo-bigrams

Every n-gram has n-1 dispersion points between its characters but we can think of it as a bigram having just one dispersion point located between a left and right part. The left part would be $c_1 \dots c_p$ and the right $c_{p-1} \dots c_n$, where $p \in \{1, p-1\}$.

This way we can calculate the *glue* of the pseudo-bigrams, assign values to n-grams and study the evolution of the glue as the size changes. The information obtained from this evolution is very important for the selection of an ngram as an MCU [3].

4.5.3 LocalMaxs

LocalMaxs is an algorithm that accepts a corpus and produces Multi Character Units (MCUs).

We define as:

Antecedent (in size) of an n-gram $c_1 \dots c_n$, ant($c_1 \dots c_n$) is a sub-ngram of $c_1 \dots c_n$ of size $n-1$ i. e. the (n-1)-gram $c_1 \dots c_{n-1}$ and $c_2 \dots c_n$

Successor (in size) of ngram $c_1 \dots c_n$, succ(M), is an $(n+1)-$gram N such as M is a ant(N), meaning that *succ(M)* contains the n-gram

For an ngram to be considered as a Multi Character Unit the following conditions must hold:
if(C's size ≥ 3)

$$g(C) \geq g(ant(C)) \wedge g(C) > g(succ(C)) \quad \forall ant(C), succ(C) \tag{4.2}$$

if(C'size = 2)

$$(C) > g(succ(C)) \quad \forall succ(C) \tag{4.3}$$

4.5.4 Measuring the *glue* holding a pseudo-bigram together

Many statistic based measures have been introduced in literature for extracting bigrams. All these measures calculate the *glue* holding the two parts of the bigram together.

Pointwise Mutual Information

Mutual Information is a symmetric, non-negative measure of common information in two variables. People thus often think of mutual information as a measure of dependence between two variables. It is measured as:

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x) \cdot p(y)}$$

For measuring the *glue* of bigrams, *pointwise mutual information* between two particular points in their distributions is calculated as:

$$I(x,y) = \log \frac{p(x,y)}{p(x) \cdot p(y)} \tag{4.4}$$

If one considers the two way contingency table of a n-gram x and n-gram y where A is the number of times x and y are found as a bigram, B is the number of times x occurs without y, C is the number of times y occurs without x and N is the total number of documents then mutual information can be estimated using:

$$I(x,y) \approx \log \frac{A \times N}{(A+C) \times (A+B)}$$

$I(x,y)$ has a natural value of 0 if x and y never occur together.

A problem with pointwise mutual information is that it does not work very well for low frequency events. Sparseness is a particularly difficult problem for mutual information. To see why, we should notice that mutual information is a log likelihood ration of the probability of the bigram $P(xy)$ and the product of the probabilities of the individual n-grams x and y ($P(x) \times P(y)$. We shall examine two extreme cases:

1. For perfect dependence we have:

$$I(x,y) = \log \frac{P(xy)}{P(x) \cdot P(y)} = \log \frac{P(x)}{P(x) \cdot P(y)} = \log \frac{1}{P(y)}$$

   Therefore, among perfectly dependent bigrams, as they get rarer, their pointwise mutual information **increases**.

2. For perfect independence we have.

$$I(x,y) = \log \frac{P(xy)}{P(x) \cdot P(y)} = \log \frac{P(x) \cdot P(y)}{P(x) \cdot P(y)} = \log 1 = 0$$

Thus, we can conclude that mutual information is a good measure of independence. Values close to 0 indicate independence (independent of frequency). But it is a bad measure of dependence because then the score depends on the frequency of the individual n-grams. Other things being equal, pseudo-bigrams composed of low-frequency n-grams will receive a higher score than pseudo-bigrams composed of high-frequency n-grams.

One solution that has been proposed for this is to use a threshold and only look at n-grams of frequency at least 3 but this does not solve the problem. For the above reasons pointwise mutual information does not seem as a good measure for the task.

Hodges et al. [29] redefined pointwise mutual information as:

$$C(xy) \times I(xy) \tag{4.5}$$

where $C(xy)$ is the number of times the pseudo-bigram occurs to compensate for the bias of the original definition in favor of the low-frequency events.

Mutual Information has been used many times in Statistical NLP, such as for clustering words. It also turns up in word sense disambiguation.

The ϕ^2 measure

The ϕ^2 *coefficient* was introduced by (Gale & Church [30]) and has been widely used e.g. (Dunning [31]. Assuming we want to measure the glue of

a bigram we consider the following contingency table: Where $f(x, y)$ repre-

Table 4.1: Contingency table for the observed counts of each bigram

	Distribution of Y when Y is present	Distribution of Y when X is not present
Distribution of X when Y is present	$f(x, y)$	$f(\neg x, y)$
Distribution of X when Y is not present	$f(x, \neg y)$	$f(\neg x, \neg y)$

sents the absolute frequency of the bigram in which the first word is word x and the second word is y; $f(\neg x, y)$ represents the absolute frequency of the bigram in which the first word is not word x and the second is word y; etc..

Considering this contingency table, we can apply the ϕ^2 *coefficient*.

$$\phi^2((x, y)) = \frac{\left(f(x, y) \cdot N - f(x) \cdot f(y)\right)^2}{f(x) \cdot f(y) \cdot (N - f(x)) \cdot (N - f(y))}$$

Where $f(x)$ and $f(y)$ are the absolute frequencies of the 1-grams x and y. N is the number of words in the corpus

The *Loglike* measure

The *Loglike coefficient* was introduced by Dunning [31]. In Dunning's work, the detection of *composite terms* is made by applying the likelihood ratio, phrasing the null hypothesis that x and y are independent as $p(x|y) = p(x|\neg Y) = p(x)$ and using the binomial distribution.

$$Loglike((x, y)) = 2 \cdot (\log l(p1, k1, n1) + \log l(p2, k2, n2) - \log l(p, k1, n1) - logl(p, k2, n2))$$

where

$$\begin{aligned}
\log l(P,K,M) &= K \cdot \ln(P) + (M-K).\ln(1-P) \\
k1 &= f(x,y) \\
k2 &= f(x,\neg y) = f(x,\neg y) = f(x) - k1 \\
n1 &= f(y) \\
n2 &= N - n1 \\
p1 &= p(x|y) = \frac{k1}{n1} \\
p2 &= p(x|\neg y) = \frac{k2}{n2} \\
p &= p(x) = \frac{k1+k2}{N}
\end{aligned}$$

N is the number of words in the *corpus*

The Dice measure

The *Dice coefficient* (Dice [32]) is also widely used [33]. This measure of correlation is defined as:

$$Dice(x,y) = \frac{2 \cdot f(x,y)}{f(x) \cdot f(y)}$$

Symmetrical Conditional Probability Measure

SCP tests the correlation between the left (x) and the right (y) part of an ngram by taking the conditional probabilities or each one given the other and multiplying both terms.

$$SCP(x,y) = p(x|y) \cdot p(y|x) = \frac{p(x,y)}{p(y)} \cdot \frac{p(x,y)}{p(x)} = \frac{p(x,y)^2}{p(x) \cdot p(y)} \tag{4.6}$$

Generalization for ngrams

For a pseudo-bigram $c_1 \ldots c_{n-1}, c_n$ the dispersion point would be located between $c_1 \ldots c_{n-1}$ and c_n. Its SCP then can be calculated as:

$$SCP\Big((c_1 \ldots c_{n-1}), c_n\Big) = \frac{p(c_1 \ldots c_n)^2}{p(c_1 \ldots c_{n-1}) \cdot p(c_n)} \tag{4.7}$$

Fair dispersion point normalization

Using equation 4.7 to measure the n-gram's glue would naturally produce a different value for every different dispersion point we choose. For example we expect to get a different SCP value if we chose bigrams $c_1 \dots c_{n-2}$ and $c_{n-1}c_n$ than the value we would get if we chose bigrams $c_1 \dots c_{n-1}$ and c_n. To solve this problem Ferreira et al. proposed the *fair dispersion point normalization* or simply *fair dispersion*. They calculate the arithmetic average of the products determined by each dispersion point along the n-gram this way they we can have a fair measure of the n-grams glue as if the n-gram was made of a left and a right part determined by a virtual *fair dispersion point* reflecting the whole n-gram's *glue*.

$$Avp = \frac{1}{n-1} \cdot \sum_{i=1}^{i=n-1} p(c_1 \dots c_i) \cdot p(c_{i+1} \dots c_n) \tag{4.8}$$

Fair SCP

Applying the *fair dispersion concept* we get the measure defined by Silva et al. as the *fair SCP* measure.

$$SCP_{fair}(c_i \dots c_n) = \frac{p(c_i \dots c_n)^2}{Avp} \tag{4.9}$$

Avp is defined in 4.8

4.5.5 Adopting LocalMaxs to handle MCU's

We will use only 3, 4 and 5grams in our feature set. We thus have to adopt *LocalMaxs* to our needs. LocalMaxs compares every n-gram with its antecedents and successors and keeps an n-gram if it satisfies rules 4.2 and 4.3.

We can not enforce these rules to 3grams and 5grams. The algorithm compares each n-gram's glue measure to its antecedents and successors.

Trigrams: The algorithm compares each trigram with all the 4grams containing it and all the bigrams contained in it. If it finds a bigram contained in it with a greater amount of glue then it would remove the 3gram. We are not keeping any bigrams in our feature set and we can not allow the removal of a trigram in this case.

A trigram should only be removed if a 4gram that contains it, is found to have a greater amount of glue. For 3grams then rule 4.2 should not be enforced, instead of it we use the following:

A 3gram is considered to be an MCU iff:

$$g(C) \geq dec(c), \forall dec(C) \tag{4.10}$$

This way we avoid removing 3grams that have better antecedents.

5-grams: A 5gram's fair SCP measure is compared by the algorithm with the same measure of 6grams that contain it and 4grams contained in it. Since 6grams will not be included in the feature set we can not allow for any 5grams to be removed from the feature set because a better 6gram has been discovered. In the case of 5grams condition 4.2 changes to:

A 5gram is considered to be an MCU iff;

$$g(C) \geq anc(c), \forall anc(C) \tag{4.11}$$

4-grams: The only size n-grams that will be compared to both their antecedent and successors will then be the 4grams.

Enforcing these rules makes our algorithm biased to selecting more 5grams and 3grams than 4grams.

Words

To investigate the importance of n-grams of length greater than 5 without a big increase in the dimensionality of the feature space we opted to ex-

periment adding words of size 6 to 11 to the features selected. For these experiments we adopt the algorithm as follows:

- Since we will be using words of size 6 and thus some 5-grams that otherwise would be selected may have a successor more *important* we can now enforce rule 4.2 for 5grams.
- We are not selecting any words or n-grams of size greater than 11 so rule 4.11 has to be enforced for words of size 11.

Chapter 5

Experiments - Results

Any proposed new feature selection method has to be compared to the dominant to date method. As mentioned in section 4.1 Information Gain has been found to be the most effective method for feature selection for Text Categorization.

To test the effectiveness of the proposed method we had to compare the results produced when features are selected using it, to the results produced when Information Gain is used.

For this purpose we have conducted the following experiments:

Information Gain

- Select various sizes of fixed length n-grams as well as variable length n-grams.

LocalMaxs

- Select variable length n-grams (lengths 3 to 5).
- Select variable length n-grams (lengths 3 to 5) and variable length words (lengths 6 to 11).
- Select variable length n-grams after having preprocessed the texts.

To avoid the confusion caused by carriage returns in the output, without loosing the additional stylistic information they carry, we have replaced carriage returns with a special character.

To test the efficiency of the proposed method against the efficiency of information gain we use SVMs to test the classification accuracy of classifiers built using features selected by both methods.

Our training and test sets contain documents from Reuters Corpus Volume 1 (discussed in section 5.1. The documents belong to 50 authors, and both sets contain 50 documents of each one of the authors; a sum of 2500 documents in our training set and 2500 documents in our test set.

5.1 Corpus

In 2000, a large corpus for the English language, the Reuters Corpus Volume 1 (RCV1) including over 800,000 newswire stories, become available for research purposes.

A natural application of this corpus is to be used as test bed for topic-based text categorization tasks (Lewis, 2004 [27]) since each document has been manually classified into a series of topic codes (together with industry codes and region codes).

There are four main topic classes: CCAT (corporate/industrial), ECAT (economics), GCAT (government/social), and MCAT (markets). Each of these main topics has many subtopics and a document may belong to a subset of these subtopics. Although, not particularly designed for evaluating author identification approaches, the RCV1 corpus contains 'by-lines' in many documents indicating authorship. In particular, there are 109,433 texts with indicated authorship and 2,361 different authors in total.

RCV1 texts are short (approximately 2 KBytes - 8KBytes), so they resemble a real-world author identification task where only short text samples per author may be available. Moreover, all the texts belong to the same text

genre (newswire stories), so the genre factor is reduced in distinguishing among the texts. On the other hand, there are many duplicates (exactly the same or plagiarized texts). The application of R-measure to the RCV1 text samples has revealed a list of 27,754 duplicates (Khmelev and Teahan, 2003 [25]).

The RCV1 corpus has already been used in author identification experiments. In (Khmelev and Teahan, 2003 [25]) the top 50 authors (with respect to total size of articles) were selected. Moreover, in the framework of the AuthorID project, the top 114 authors of RCV1 with at least 200 available text samples were selected (Madigan, et al., 2005 [26]). In contrast to these approaches, in this study, the criterion for selecting the authors was the topic of the available text samples. Hence, after removing all duplicate texts found using R-measure, the top 50 authors of texts labeled with at least one subtopic of the class CCAT (corporate/industrial) were selected. That way, it is attempted to minimize the topic factor in distinguishing among the texts. Therefore, since steps to reduce the impact of genre have been taken, it is to be hoped that authorship differences will be a more significant factor in differentiating the texts. Consequently, it is more difficult to distinguish among authors when all the text samples deal with similar topics rather than when some authors deal mainly with economics, others with foreign affairs etc. The training corpus consists of 2,500 texts (50 per author) and the test corpus includes other 2,500 texts (50 per author) non-overlapping with the training texts.

5.2 Information Gain

To test the efficiency of our method we had to compare it to the dominant to date method for feature selection for the task of text classification. We used Information Gain to select variable length n-grams as well as fixed length n-grams. We built the following feature sets.

1. Out of the 15000 most frequent 3-grams we used Info Gain to select 2000 to 10000 3-grams (steps of 1000).

2. Out of the 15000 most frequent 4-grams Info Gain selected 2000 to 10000 (steps of 1000).

3. Out of the 15000 most frequent 5-grams Info Gain selected 2000 to 10000(steps of 1000).

4. Out of the 15000 most frequent variable length n-grams (5000 3-grams, 5000 4-grams, 5000 5-grams) Info Gain selected 2000 to 10000 with a step of 1000.

We used these four sets of features to build classifiers and tested their efficiency by classifying the documents in our test set.

The results of these tests are depicted in table 5.1 and figure 5.1.

As can bee seen Info Gain does not perform well when small numbers of features are selected (up to 3000 n-grams), even though these 3000 n-grams have been selected out of the 15000 most frequent n-grams.

Results improve for all sizes of fixed length as well as for the variable length n-grams until the number of n-grams selected goes up to 6000, results then do not have any noticeable peaks or drops up to 10000 n-grams.

3-grams do not perform well for small numbers of selected features (2000-4000) but their performance increases when their number reaches 6000 and then it gets better results than 5-grams and variable length n-grams.

The best overall results as well as the highest result of **73.92**% where achieved when using n-grams of size 4.

Info Gain ranks all the n-grams but does not append a ranking value to all of them.

Table 5.1: Results with features selected using the Information Gain measure

features	3grams	4grams	5grams	variable length n-grams
2000	65.44	66.72	67.28	67.23
3000	65.64	68.68	69.24	70.76
4000	66.48	71.04	71.80	72.25
5000	70.68	72.32	71.82	72.32
6000	73.32	73.16	72.12	72.52
7000	73.44	73.72	72.56	73.04
8000	73.84	73.48	72.80	72.52
9000	73.80	73.28	72.32	72.76
10000	73.72	73.92	72.56	72.76

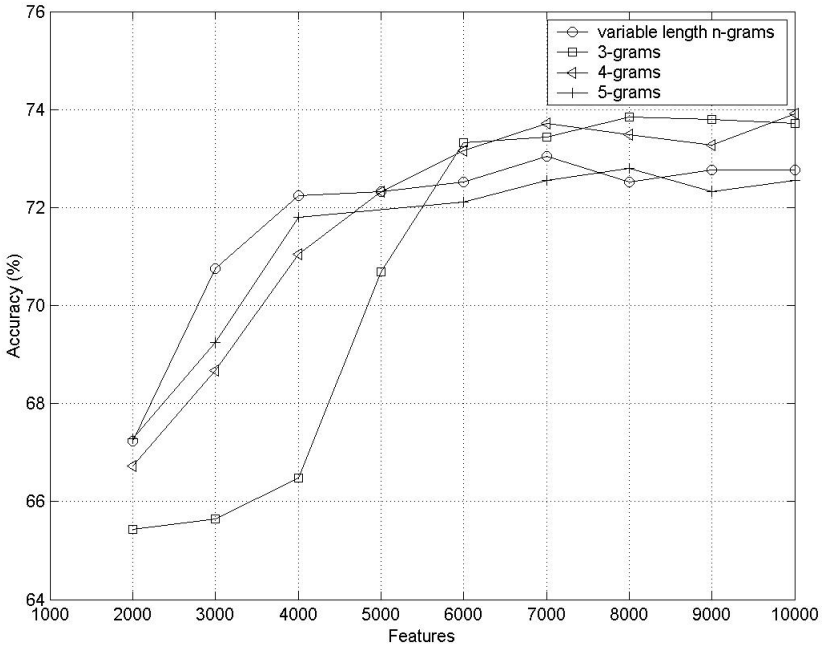

Figure 5.1: Results with IG selected features

Table 5.2: Numbers of features selected by LocalMaxs using different sizes of initial feature sets

Initial feature set			Numbers of n-grams selected			
3grams	4grams	5grams	3grams	4grams	5grams	Total
2000	2000	2000	1337	423	554	2314
3000	3000	3000	2130	564	822	3516
4000	4000	4000	2939	705	1049	4692
5000	5000	5000	3786	821	1254	5861
6000	6000	6000	4656	910	1448	7014
7000	7000	7000	5510	1012	1656	8178
8000	8000	8000	6362	1111	1847	9320

5.3 LocalMaxs

5.3.1 Variable length n-grams

LocalMaxs calculates the glue that holds the characters of an n-gram together, compares the glue values of each n-gram with the respective measure of its antecedents and successors and keeps the dominant ones.

LocalMaxs does not produce a ranking of the n-grams, therefore we have no way of controlling the number of n-grams it extracts. However we can produce different numbers of n-grams by selecting them from different initial feature sets. Table 5.2 depicts the numbers of n-grams selected by LocalMaxs from different size initial sets of features [1].

As expected from all sizes of initial feature sets the algorithm favors the selection or 3-grams and 5-grams over 4-grams.

For our first experiments using LocalMaxs we used the 3000 to 24000 most frequent variable length n-grams (with a step of 1000 per size as can

[1]The initial feature sets are selected ranked by frequency

Table 5.3: Results using LocalMaxs

features	Accuracy (%)
1134	68.56
2314	**72.00**
3516	71.88
4692	72.48
5861	73.08
7014	73.64
8178	**74.04**
9320	73.76

be seen in table 5.2). The results of those experiments can be seen in table 5.3.1 and figure 5.2.

The highest accuracy of **74.04**% was reached when 8,178 n-grams where selected from an initial feature set of the 21,000 most frequent variable-length n-grams [2].

LocalMaxs selects good enough features to reach an accuracy of **72**% (table 5.3.1) using only 2314 n-grams, selected out of the 6000 most frequent variable length n-grams.

[2] 7,000 3-grams, 7,000 4-grams and 7,000 5-grams

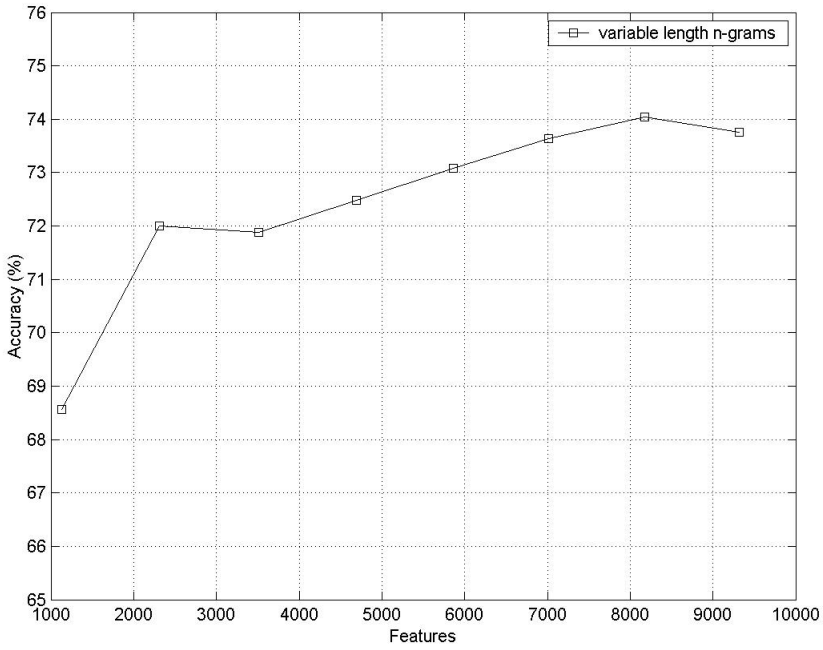

Figure 5.2: Results LocalMaxs

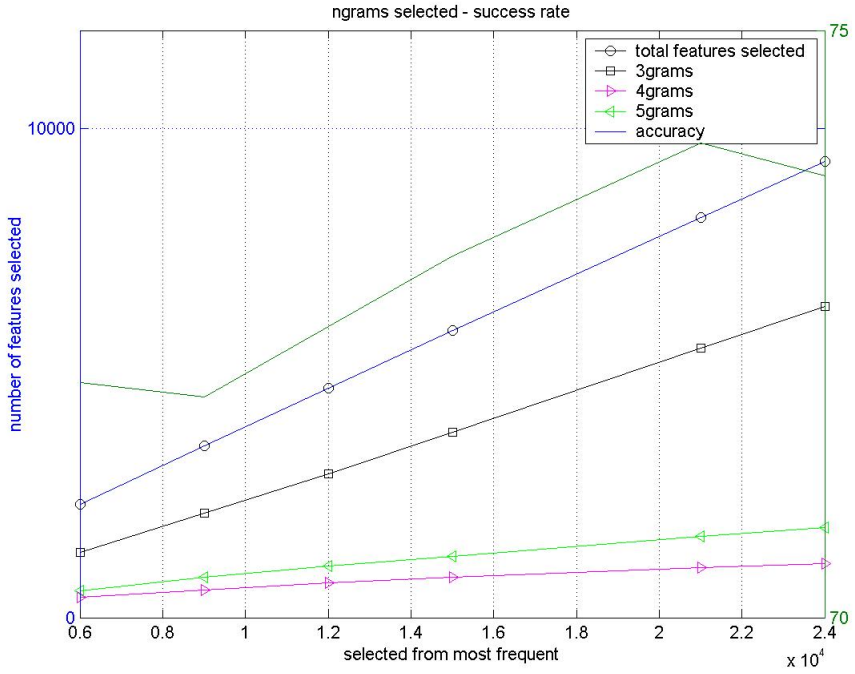

Figure 5.3: Features Selected by LocalMaxs and Accuracy of the classifiers produced

Figure 5.3 depicts how the accuracy of the classifiers develops as the number of features selected increases.

Exploring the differences of the Feature sets Selected

In this section we compare the feature sets produced by information gain and LocalMaxs. We compare the sets of variable length n-grams produced by IG from raw text to the sets of n-grams produced by LocalMaxs from raw text.

Common n-grams

LocalMaxs selects a set of features with very small similarity to the one IG does. Table 5.4 shows that when 2314 variable length n-grams are selected

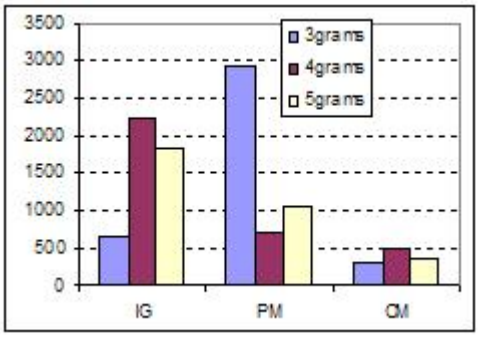

Figure 5.4: Similarity of feature sets selected by infogain and LocalMaxs for the case both methods select 4,691 features

Table 5.4: Common n-grams

	IG	PM	Com	IG	PM	Com	IG	PM	Com
3-grams	647	1337	127	647	2938	317	851	5510	530
4-grams	909	423	161	2228	705	462	2327	1012	510
5-grams	758	554	131	1816	1048	315	5000	1656	1257
Total	2314	2314	419	4691	4691	1094	8178	8178	2297
Accuracy	69.4	72.00		72.16	72.48		72.56	74.04	

using IG (out of 15000) and LocalMaxs (out of 6000) the common 3-grams picked are only 127 and there is a **2.6**% difference in accuracy, proving that the n-grams selected belong to a completely different set. Obviously the n-grams selected by LocalMaxs are better able to distinguish the stylistic differences of each Author's writing.

As the number of n-grams selected increases the algorithm is biased towards selecting more 3grams. On the other hand information gain tends to select more 5-grams.

Ranking (by Info Gain) of the n-grams selected

Info Gain Figure 5.5 depicts the distribution of variable length n-grams picked by Info Gain as they where ranked by it.

To explain the results depicted in figure 5.5 let us examine the subplot labeled *Figure 3 4000 features*[3]. It is the case 4000 variable n-grams are selected using info gain.

The y-axis represents the number of features selected. The x-axis represents the ranking of the features selected; for example there where approximately 900 features selected that belong to the features ranked between 4000 and 5000 by Info Gain, and approximately 500 features selected that ranked between 1 and 1000.

It seems that Info Gain is biased to selecting the n-grams ranked between 5000 and 7000 and when they are finished then it selects the ones between the 1st and 4th thousand.

The same bias is displayed with fixed length n-grams as depicted in figures 5.6, 5.7 and 5.8.

[3]On the upper right corner of the figure

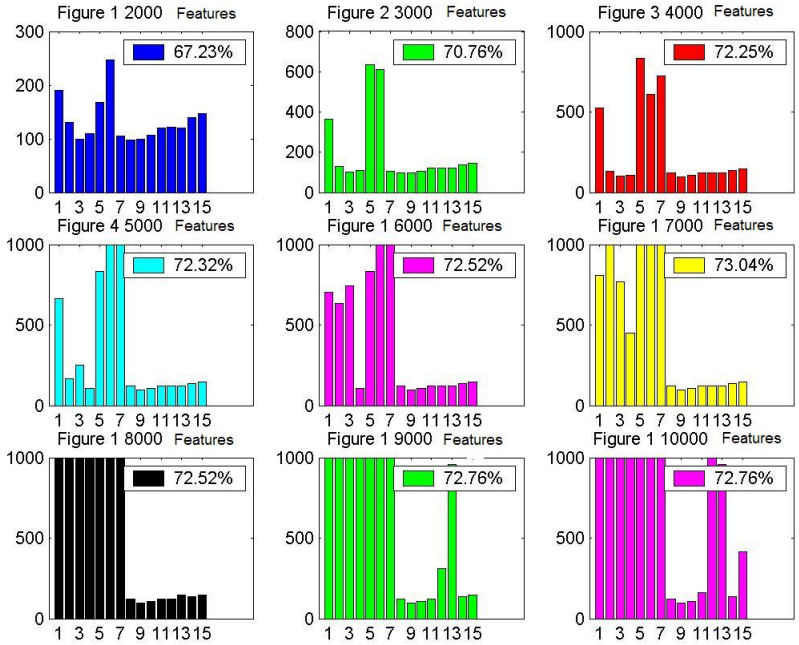

Figure 5.5: Variable length n-grams picked by Info Gain and their distribution by frequency.

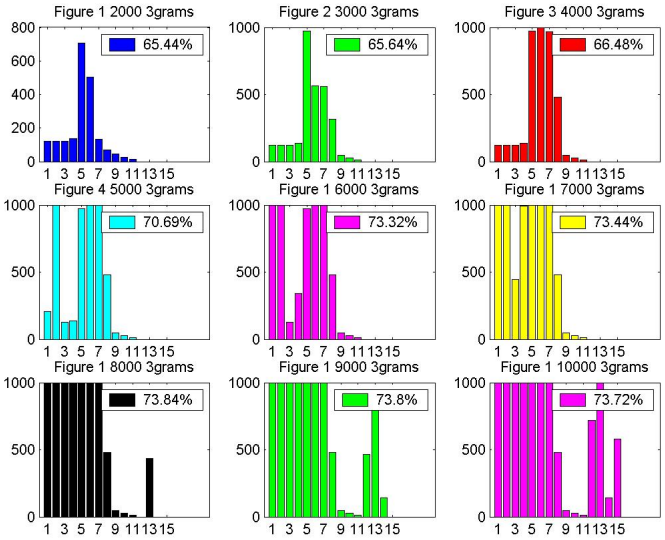

Figure 5.6: 3-grams picked by Info Gain and their distribution by frequency.

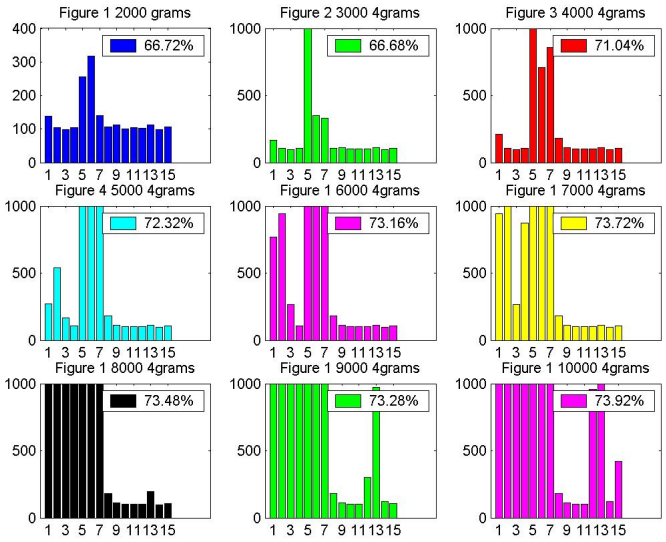

Figure 5.7: 4-grams picked by Info Gain and their distribution by frequency.

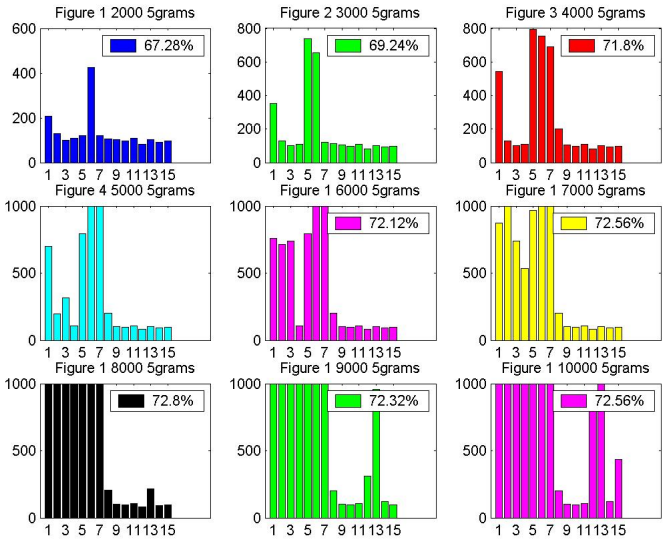

Figure 5.8: 5-grams picked by Info Gain and their distribution by frequency.

LocalMaxs Figure 5.9 depicts the distribution of variable length n-grams picked by LocalMaxs as they rank by frequency when it selects 5861 n-grams out of the 15,000 most frequent.

As can be seen LocalMaxs selects mostly 3-grams that ranked low by frequency. This is a pattern that occurs every time the algorithm is used.

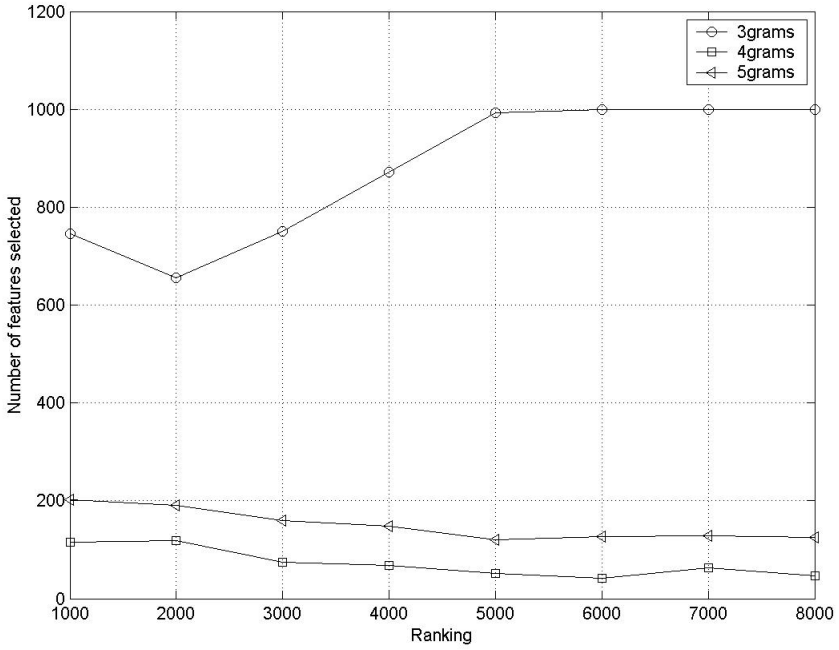

Figure 5.9: Variable length n-grams picked by LocalMaxs from an initial feature set of 15,000 most frequent n-grams and their distribution by frequency.

LocalMaxs selects all of the n-grams that have ranked low by frequency.

The 4-grams and 5-grams selected follow a pattern easier to explain as more n-grams are selected among the high ranking n-grams than the ones selected from low ranking n-grams.

5.3.2 Pre processed text

The experiments we have presented so far were conducted on raw text. No pre-processing of the text was performed apart from removing XML tags irrelevant to the text itself.

However, simple text pre-processing may be helpful in the framework of author identification tasks.

Digits used in text do offer stylistic information that can be used for authorship classification. However the information carried by digits in the case of the task at hand is not contained in the different combinations of digits used but the actual use of digits in the text. Digits are discussed in section 3.5.1.

Since we are not interested in the different combinations of digits we can replace digits with character "@". This would affect the number of n-grams produced as well as the n-grams themselves.

For example numbers |1233|, |1932|, |3284| and all different combinations of 4 digits appearing in a text just once would receive a different glue value than |@@@@|, a string of 4 symbols appearing many times in the text [4].

We examine the effect of this simple pre-processing procedure on the authorship identification task. Figure 5.10 depicts the classification accuracy results using the proposed feature selection method on variable-length n-grams extracted from raw text (as previously) and pre-processed text (with digit characters replaced by a symbol). As can be seen, the numbers of features selected based on the pre-processed text are slightly smaller. More

[4] If digits are replaced with it.

Table 5.5: Results on pre-processed text

n-grams	Accuracy
1130	68.72%
2300	71.92%
3513	72.40%
4698	72.88%
5867	73.16%
7018	74.00%
8168	74.16%
9333	74.36%

importantly, the performance of the model based on pre-processed text is better especially when using more than 2,000 features. This indicates that considerable improvement in accuracy can be achieved by simple text transformations.

Replacing digits with the character "@" did help LocalMaxs select features that raise substantially the results of Authorship Identification.

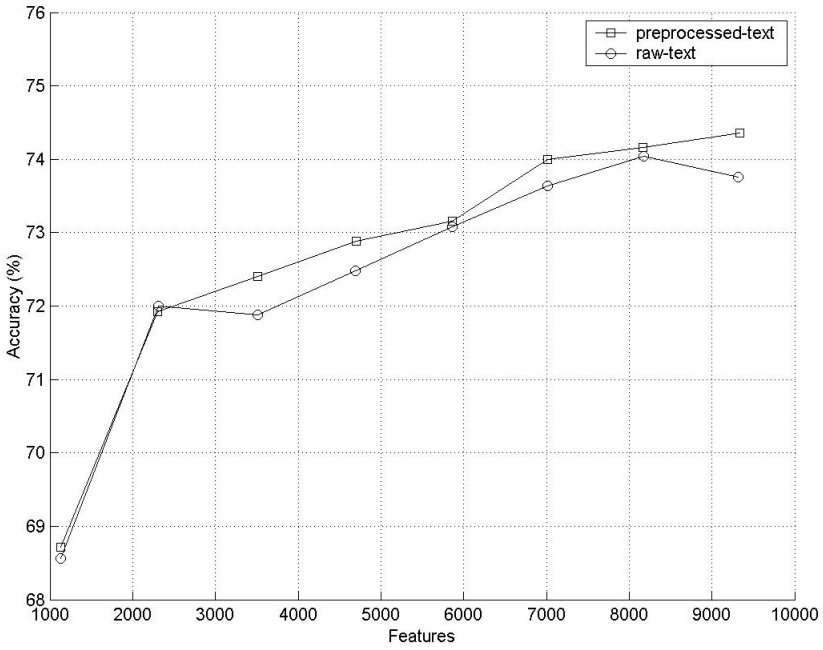

Figure 5.10: Results pre-processed text vs raw text

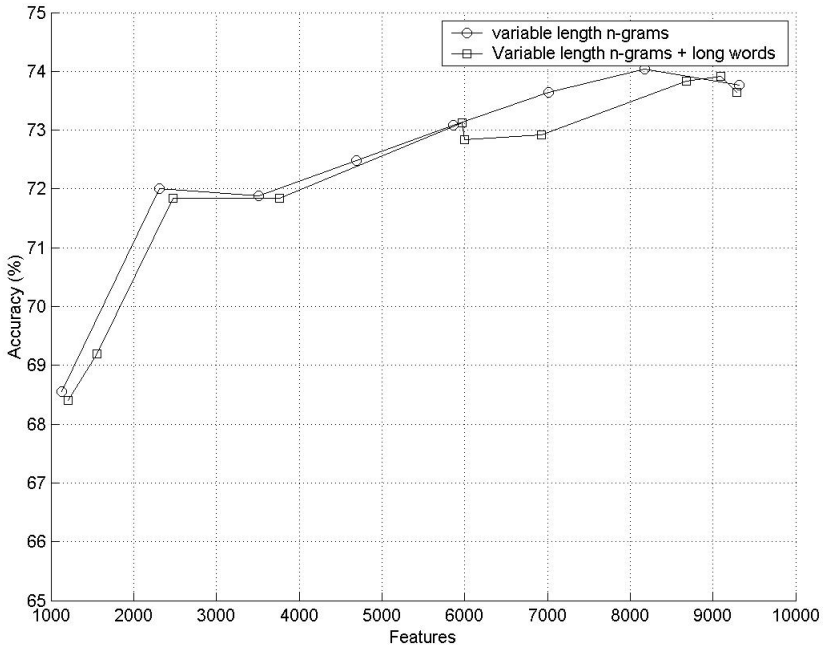

Figure 5.11: Variable Length n-gram + long words vs Variable length n-grams

5.3.3 Variable length n-grams + long words

To explore the effectiveness of our method when n-grams of higher order are used and to avoid the uncontrollable increase in the size of the feature set we added words to the feature set. We selected the most frequent 6 to 11 character words and added them to the features.

The results of these tests are depicted in figure 5.3.3 and table 5.6.

As it can be seen words did not have much to offer to the categorization task. Results did not improve and in some cases they dropped.

Table 5.6: Results using variable length n-grams + long words

n-grams + words	Accuracy
1204	68.40%
1561	69.20%
2479	71.84%
3760	71.84%
5972	73.12%
6931	72.92%
8679	73.84%
9093	73.92%
9285	73.64%

5.4 Time Efficiency

On a Pentium M 1.73GHz the proposed method needs 4 minutes and 20 seconds (**260 seconds**) to select variable length n-grams directly from the training set. Time is linear on the size of the training set and independent of the number of n-grams to be selected.

This is only a fraction of time that Info gain needs to process the same number of features, because as can Info gain has quadratic complexity [5].

[5]Time can be influenced by the implementation too. We are using the WEKA implementation of IG

Chapter 6

Conclusions - Future Work

In this Thesis:

- We have introduced the use of variable length n-grams for the task of authorship identification
- We have proposed a new method to feature selection for authorship identification based on character n-gram text representation.

  The proposed method is based on previous work by Silva et al. [3]. They introduced the LocalMaxs algorithm for extracting multi character units from text.
- We have explored the effect of simple preprocessing of text prior to feature selection.

To test the effectiveness of the proposed method we compared the results produced from classifiers built from feature sets produce using it, against results produced with classifiers built with features selected by info gain; the dominant feature selection method for text classification to date.

We have used SVMs to test the efficiency of our method. SVMs are well suited for the task of test categorization (section 2.2.5). We are using the *Sequential Minimal Optimization* algorithm for training support vector

machines developed by Platt [40] as it is implemented in WEKA [1] [41].

Variable Length n-grams

Work until now in the field of authorship identification has shown that the best results can be achieved using fixed length n-grams of sizes 3 to 5 [4] to represent the texts. In fact in June 2004, at the "Ad-hoc Authorship Attribution Competition" the highest scoring participant was the research group of Vlado Kešelj, with an average success rate of approximately 69% [21] using character n-grams of fixed length. Kešelj et al. have produced very good results in [4] using fixed length n-grams and got their best results using n-grams of sizes 3, 4 **or** 5.

For this reason and to keep the dimensionality of the feature space to a manageable size we opted to use variable length n-gram of sizes 3, 4 **and** 5.

The main reason variable length n-grams have not been used until now is that distinct n-grams contained in corpora of even small sizes can range to hundreds of thousands or even millions. As can be seen in table 3.2 the subset of RCV1 (section 5.1) we are using contains 457,808 distinct variable length n-grams (only of sizes 3,4 and 5).

This is a size of feature set almost impossible for any machine learning algorithm to handle. To reduce the feature set to a manageable size, we are introducing LocalMaxs as an alternative to Information Gain method of feature selection method.

LocalMaxs

LocalMaxs (section 4.5.3) measures the *glue* that holds together all the n-grams in a corpus and then it compares each n-gram with all its antecedents [2] and all its successors [3]. It keeps only the n-grams that have a

[1] The *Waikato Environment for Knowledge Analysis*

[2] n-grams of size n-1, contained in it

[3] n-grams of size n+1 that contain it

greater amount of glue holding them together than all their antecedents and successors. We had to change LocalMaxs and adjust it to produce variable length character n-grams.

Since we only use 3-grams, 4-grams and 5-grams we had to change the algorithm for the 3-grams and 5-grams and compare their glue value only to their successors and antecedents respectively.

The only n-grams that are compared to both their antecedents and successors are 4-grams. This makes the 4-grams the size that is selected the least by LocalMaxs. 4-grams though have proven to be important for the task. In [4] the best results achieved in authorship attribution of Greek and Chinese texts where using 4-grams. In our experiments with info gain (section 5.2) using fixed length n-grams as well as variable length n-grams we got our best result using 10,000 4-grams.

Considering that it has been proven in [6] that even low ranking features hold valuable information for the task of text categorization, we should keep an open matter the probability of partly relaxing the LocalMaxs rules in future work, and try again with some more 4-grams in our feature set.

With the present settings the most 4-grams used in our feature set are 1,111 4-grams when 9,320 n-grams are selected from the 24,000 most frequent n-grams.

As it was expected by the nature of LocalMaxs; the feature sets selected were different then the ones selected by information gain.

For example when both methods are used to select 8,178 variable length n-grams, IG selects 851 3-grams and the PM [4] selects 5,510 3-grams, of these only 530 are common in both sets. In the same experiment IG selected 2,327 4-grams and the PM 1,012 4-grams, of these features only 510 are common to both sets (see section 5.3.1, table 5.4).

What are the reasons LocalMaxs proves to be better than Info Gain in

[4]Proposed Method

selecting features for the task of Authorship Identification?

LocalMaxs does not just pick n-grams based on their importance for determining the category, and it does not select them based on frequency alone.

Frequency does affect the glue value but alone is not enough for an ngram to get selected.

- N-gram "␣the␣" appears 61967 times but is not selected as an MCU.
- "The" appears 10837 times and is selected receives a high glue score.
- "␣and␣" appears 25605 times - not selected.
- "And" appears 275 times - selected.

When LocalMaxs keeps 5861 n-grams it only selects 8 n-grams containing "and", leaving room for other *important* n-grams in the feature set. It uses frequency to calculate the *glue* that holds the characters together but it also compares it with its antecedents and successors and will keep an n-gram iff its glue measure is greater than theirs.

Therefore, the produced feature set is stylistically richer since it contains the dominant character n-grams and is less likely to be biased on some powerful n-grams that essentially represent the same stylistic information.

An interesting fact that is that when it comes to 3-grams the proposed method favors the selection of low frequency n-grams. As can be seen in 5.9, from an initial set of 15,000 n-grams, LocalMaxs selects 655 3-grams that ranked [5] between 1,000 and 2,000 and all of the 3-grams ranked between 4,000 and 8,000. This is something that will have to be investigated further in future work.

Another difference with traditional feature selection approaches is that there is no ranking of the features according to their significance. This fact does not allow the selection of predefined numbers of features.

[5] by frequency

This fact however only affects experimental comparisons with other approaches rather the practical application of the proposed method to real-world cases.

In this study, we restricted our method to certain n-gram types (3-grams, 4-grams, and 5-grams). To keep dimensionality on low level, we used word longer than 5 characters as an alternative for long n-grams. However, the results when using the additional words were not encouraging. It would be interesting for one to explore the full use of long n-grams as well as the distribution of selected n-grams into different n-gram lengths especially when texts from different natural languages are tested.

Text preprocessing

We have presented experiments exploring the significance of digits as stylistic features in the framework of author identification tasks. The removal of redundancy in digit characters improves classification accuracy when a character n-gram text representation is used. Furthermore, the cost of this procedure is trivial. It remains to be tested whether alternative text transformations are useful as well.

''n-grams'' software

A valuable tool for future work has been developed for the purposes of this thesis. **''n-grams''** is a software package with a graphical user interface front. It contains classes for many text processing algorithms and features. It can select features (words and n-grams of variable or fixed length) using frequency, LocalMaxs and SCP or Pointwise Mutual Information, as well as term frequency thresholding. For all those feature selection tasks it produces statistics useful for analyzing the feature sets produced. It contains batch text concatenating features as well as fast implementation of a powerful regular expression search tool **egrep** and a regular expression testing

interface.

It also contains classes for specific to WEKA users needs, as for producing arff files, processing Info gain feature selection files etc.

Closing Comments

It has been shown that variable length n-grams represent stylistic features better than fixed length n-grams. LocalMaxs also has been found to be at least as good a feature selection method as info gain is and in many cases better than it. Future work should focus on using a wider set of variable length n-grams as well as fine tuning the LocalMaxs algorithm and testing different statistic measures for the glue holding pseudo-bigrams together.

In all our tests 74% was something of an upper limit in accuracy. The purpose of this thesis was not to test the efficiency of SVMs. Future work should also focus on testing different machine learning algorithms. Kešelj et al. achieved accuracy of over 90% using the profile based approach to author classification and a simple way to measure the distance of profiles. In future work we intend to use the proposed method to build profile based classifiers, as well as test it for selecting features for other text categorization tasks, as spam identification and program code author identification.

Bibliography

[1] Stamatatos,Kokkinakis,Fakotakis: *'' Automatic Text Categorization in Terms of Genre and Author''* Computational Linguistics 26:4 (2000) 471-495.

[2] Stamatatos: *Computer-Based Authorship Attribution wihtout Lexical Measures*

[3] Silva,Lopes: *''A Local Maxima method and a Fair Dispersion Normalization for extracting* ***multi-word units*** *from corpora* In Proc. of the 6th Meeting on the Mathematics of Language (1999) 369-381.

[4] Kešelj,Peng,Cercone: *''ngram Based Author Profiles for Authorship Attribution''* In Proc. of the Conference Pacific Association for Computational Linguistics (2003)

[5] Yang Y., Pedersen J.: *A comparative Study on Feature Selection in Text Categorization*,In Proc. of the 14th Int. Conf. on Machine Learning (1997)..

[6] Thorsten Joachims: *''Text Categorization with Support Vector Machines: Learning with Many Relevant Features''* European Conference on Machine Learning (ECML), 1998.

[7] Witten,Frank: *''Data Mining''* Morgan Kaugmann, Second Edition 2006

[8] Jurafsky, Martin: *''Speech and Language Processing''* Prentice Hall

[9] Manning and Schutze: *''Foundations of Statistical Natural Language Processing''*, MIT Press, Sixth printing 2003.

[10] Mosteller, Frederick and Wallace, David L. Inference and Disputed Authorship: The Federalist. 1964.

[11] Sebastiani: *A Tutorial on Automated Text Classification*

[12] Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys, 34:1 (2002) 1-47.

[13] Arnold, Gosling, Holmes: *''The Java Programming Lanquage''* Addison Wesley, Fourth Edition

[14] Tsolomitis,Syropoulos,Sofroniou: *"Digital Typography Using Latex"* Springer

[15] Olivier de Vel: *Mining E-mail Authorship*

[16] Diederich Joachim: *Authorship Attribution with Support Vector Machines*

[17] Rudman, J: *''The state of authorship attribution studies: Some problems and solutions*

[18] Yang, Y: *''A re-examination of text categorization methods''*

[19] Stamatatos: *Machine Learning - Lecture 1 Introduction*

[20] Kivinen,J:*''The perceptron algorithm vs. winnow: Linear vs. logarithmic mistake bounds when a few input variables are relevant.''* In conference on Computational Learning, McGraw-Hill, 1997.

[21] Juola, P.: *''Ad-hoc Authorship Attribution Competition''*. In Proc. of the Joint ALLC/ACH2004 Conf. (2004) 175-176.

[22] Masand B., Linoff G:*''Classifying news stories using memory based reasoning,''* In 15th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval, pages 59-64, 1992.

[23] Mitchell: *''Machine Learning''* McGraw Hill, New York 1996

[24] Ruiz M,Srinivasan P.: *'''Hierarchical neural networks for text categorization'* In Proceedings of SIGIR-99, 22nd ACM International Conference on Research and development in Information Retrieval (Berkley, CA, 1999), 281-282

[25] Khmelev, D. Teahan, W.: A Repetition Based Measure for Verification of Text Collections and for Text Categorization. In Proc. of the 26th ACM SIGIR (2003) 104-110.

[26] Madigan, D., Genkin, A., Lewis, D., Argamon, S., Fradkin, D., Ye, L.: Author Identification on the Large Scale. In Proc. of CSNA (2005).

[27] Lewis, D., Yang, Y., Rose, T., Li, F.: RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research, 5 (2004) 361-397.

[28] Kessler B., Nunberg G.,Schutze: *''Automatic Detection of Genre''*

[29] Hodges et al.*''An automated system that assists in the generation of document indexes ''*. Natural Language Engineering 2:137-176

[30] Gale,W.Church,K: *''Concordance for parallel texts''* in Proceedings of the Seventh Annual Conference of the UW Center of the

new OED and Text Research, Using Corpora, pp.40-62, Oxford, 1991

[31] Dunning, T: *Accurate methods for the Statistic of Surprise and Coincidence* Association for Computational Linguistics, 19(1): 61-76 1993.

[32] Dice L: *''Measures of the Amount of Ecologic Association between species''*, Journal of Ecology, 26:297-302

[33] Smadja F,Hatzivassiloglou: *''Translating Collocations for Bilingual Lexicons: A Statistical Approach''*. Assosiation for Computational Linguistics.

[34] Lodhi et al.: *Text Classification using String Kernels*

[35] Spafford E., Weeber S: *''Software forensics: tracking code to its authors''* Computers and Security, 12:585-595, 1993

[36] Wang et al. *''Generating Concept Hierarchies from Text for Intelligence Analysis''*

[37] Eyheramendy S, Lewis D, Madigan D: *''On the Naive Bayes Model for Text Categorization''* Proceedings Artificial Intelligence & Statistics 2003

[38] McCallum A, Nigam K: *''A Comparison of event models for Naive Bayes Text Classification''* Proceedings fo AAAI-98 Workshop on *Learning for Text Categorization*, 1998

[39] Peng et al.: *''Augmenting Naive Bayes Classifiers with Statistical Language Models''*

[40] Platt John: *''Sequential Minimal Optimal: A fast Algorithm for training Support Vector Machines''*

[41] Web: http:

www.cs.waikato.ac.nz/ml/weka

Index