Modeling Non-Standard Text Classification Tasks

Faculty of Media
Bauhaus-Universität Weimar
Germany

Dissertation of
Nedim Lipka

To obtain the academic degree of
Dr. rer. nat.

Advisor:	Prof. Dr. Benno Stein
Reviewer:	Dr. James G. Shanahan

Contents

Abstract v

Notation vii

1 Introduction 1
1.1 Thesis structure . . . 3
1.2 Contributions . . . 6

2 The history of text classification 11
2.1 Who is speaking . . . 12
2.2 What is the text about . . . 15
2.3 To whom is the text addressed . . . 16
2.4 Which channel is used . . . 17
2.5 What is the effect of the text . . . 17
2.6 A comprehensive task: Web genre analysis . . . 19

3 Supervised learning 23
3.1 An explanatory model of generalization: inductive bias . . . 23
3.2 Inductive biases of common learning algorithms . . . 28
3.3 Optimization for learning algorithms . . . 29
3.4 Evaluation of text classification solutions . . . 34

4 Feature engineering for non-standard text classification tasks 37
4.1 Co-stems in non-standard text classification tasks . . . 38
4.2 Core-vocabularies in Web genre analyses . . . 45
4.3 *N*-grams in information quality analyses . . . 55
4.4 *N*-grams in short text language identification . . . 59
4.5 Function words in authorship verification analyses . . . 63
4.6 Style markers in intrinsic plagiarism analyses . . . 67
4.7 Bibliography . . . 79

5 Towards effective text classification in the wild 83
5.1 Improving generalization capabilities via robust feature engineering 84
5.2 Sampling informative examples via advanced active learning . . 89
5.3 Learning target classes via one-class classification . . . 95
5.4 Improving one-class classification via ensembles . . . 100
5.5 Bibliography . . . 104

6 Model selection for text classification in the wild 107
6.1 Expected effectiveness under subclass distribution shifts 109
6.2 An expected effectiveness estimate 110
6.3 An expected effectiveness heuristic 111
6.4 Model selection via expected effectiveness 114
6.5 Validation of prior assumptions 115
6.6 Application and conclusions 118
6.7 Bibliography 120

7 Conclusions 123

References 125

Abstract

Text classification deals with discovering knowledge in texts and is used for extracting, filtering, or retrieving information in streams and collections. The discovery of knowledge is operationalized by modeling text classification tasks, which is mainly a human-driven engineering process. The outcome of this process, a text classification model, is used to inductively learn a text classification solution from a priori classified examples. The building blocks of modeling text classification tasks cover four aspects: (1) the way examples are represented, (2) the way examples are selected, (3) the way classifiers learn from examples, and (4) the way models are selected.

This thesis proposes methods that improve the prediction quality of text classification solutions for unseen examples, especially for non-standard tasks where standard models do not fit. The original contributions are related to the aforementioned building blocks: (1) Several topic-orthogonal text representations are studied in the context of non-standard tasks and a new representation, co-stems, is introduced. (2) A new active learning strategy is examined that goes beyond standard sampling. (3) A new one-class ensemble for improving the effectiveness of one-class classification is proposed. (4) A new model selection framework is introduced to cope with subclass distribution shifts that occur in dynamic environments.

Notation

Elements & Sets

D	population of documents
$d \in D$	document (an unstructured text unless otherwise mentioned)
$Y = \{+1, -1\}$	(binary) class scheme
$y \in Y$	class label
$\mathbf{D}$	population of feature vectors
$\mathbf{d} \in \mathbf{D}$	feature vector that represents the document d
$\mathbf{S} \in 2^{\mathbf{D}}$	sample of feature vectors ($2^{\mathbf{D}}$ denotes the power set of D)

Classification

$c : D \rightarrow Y$	target concept or ground truth
$\alpha : D \rightarrow \mathbf{D}$	feature engineering function
$\beta : \mathbf{D} \rightarrow \{0, 1\}$	sampling strategy
$h : \mathbf{D} \rightarrow Y$	classifier or hypothesis
$L : 2^{\mathbf{D}} \rightarrow H$	learning algorithm ($2^{\mathbf{D}}$ denotes the power set of D)
H	hypothesis space
$T = (D, Y)$	text classification task
$M = (\alpha, \beta, L)$	text classification model
$m = (\alpha, h)$	text classification solution
NB	naïve bayes classifier
SVM	support vector machine

Probability

$\mathcal{X}$	random variable that models the document distribution over D
$\mathcal{Y}$	random variable that models the class distribution over Y
$\mathcal{H}$	random variable that models the hypothesis distribution over H
E	expected value
P	probability

Effectiveness measures

Acc	accuracy
F	F-measure
P, Prec	precision
R, Rec	recall
AUC	area under ROC curve

[0] The table excludes notation that are exclusively used in single chapters.

Chapter 1

Introduction

This thesis deals with technologies for automatic text classification based on machine learning and aims to improve the effectiveness of classifying unseen texts. The focus is on non-standard text classification tasks.

We call a task “non-standard” if the classification goes beyond the texts subjects and a machine learning algorithm cannot generalize under bag-of-words models. A bag-of-words model is a common text representation that reduces texts to an unordered collection of words and utilizes the word frequencies as features. Figure 1.1 shows a variety of non-standard text classification tasks. It shows the effectivenesses of a classifier in terms of the F-measure when bag of words (stems) and a complementary representation (co-stems) are employed. To illustrate the difference of these representations consider for example the word “timelessly”, which is either represented by its stem “time” or by its co-stem “lessly”. Tasks on the diagonal in the figure perform equally well under both representations; however, the complementary representation is independent of the content of the texts and is preferred because of the resulting generality. Therefore, tasks near the diagonal are referred to as non-standard. Typical standard tasks are found in the upper left of the figure, whereby a bag-of-words model is sufficient to achieve a high effectiveness. Topic categorization, for example, classifying news articles from the 20 Newsgroups corpus by their topic, is clearly a standard task in contrast to many other tasks in the figure. Non-standard text classification tasks provide clues about in-depth knowledge such as

- authorship,
- author’s gender,
- effects (is the text humorous, an act of vandalism, spam),
- sentiment,
- genre,
- sector, or
- information quality.

Improving the effectiveness of non-standard tasks has a high impact on several applications. The information age reclaims filtering, extraction, and retrieval

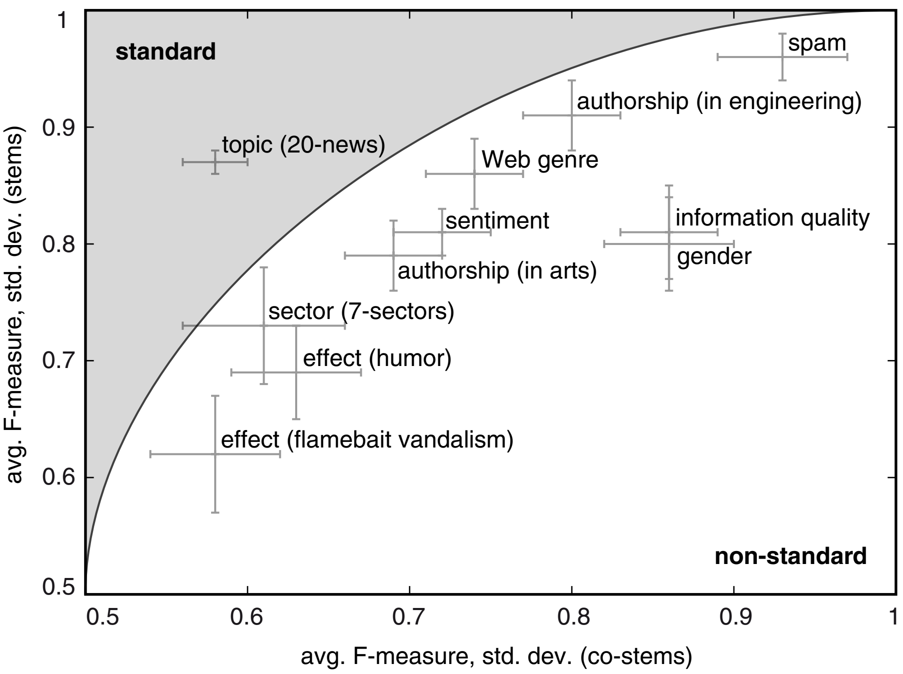


Figure 1.1: *The figure presents a landscape of text classification tasks mostly considered to be non-standard. The axes show the averaged effectiveness of a linear support vector machine classifier under different text representations: (1) a standard bag-of-words representation with word stems on the y-axis and (2) the complementary representation on the x-axis. The complementary representation is based on the residuals if the word stems are removed from the text; therefore, topic information is excluded. The co-stem of the word "timelessly" is "lessly".*

applications to satisfy the society's thirst for knowledge and to gain insights used as intermediate inputs or as final results in (big) data mining processes. These applications need to handle unstructured texts, which are a substantial source of information in the Web. Common examples are authorship and Web genre analyses. Authorship analysis is a tool for forensic linguistics, which helps to verify claims of responsibility, confessions, wills, or to identify plagiarism offenses. In particular, plagiarism identification requires a high degree of automation; the number of suspicious texts is growing since publishing in the Web and reusing content is easier than ever. Web genre analysis is applied for filtering blog, news, or scientific articles and the like; the number of these applications is growing and they are no niche products anymore. It should be noted that the results of genre analyses can also be reused as metadata to improve the ranking effectiveness of vertical search engines. This applies in general for text classification, and our leitmotif is:

> *"The more knowledge exists about a text, the higher is the potential of filtering, extracting, or retrieving it."*

While the loss of information that occurs under bag of words, namely the order of words, is not decisive, bag of words misses to unveil deeper text characteristics to the classifier. As an unintended consequence a different target concept is learned; classifiers with a bag-of-words representation are likely to become topic-, domain-, or time-dependent and do not generalize beyond these bounds. For example, a classifier that is trained to distinguish between authors through a bag-of-words lens implicitly learns to distinguish between the topics in the training sample (texts that are given to the learning algorithm) instead of unveiling the authors' writing styles.

In addition to the employed model of text representation, the training sample takes an important role and decides on the appropriateness of learning algorithms. A training sample has to be representative for the population of texts and the target concept. The representativeness is compromised by the following aspects, which also characterize non-standard tasks:

- lack of target or complementary examples and open classes
- unknown class balances and drifting target concepts

If only the target class is properly represented in the training sample but not the complementary class, a standard learning algorithm is not able to evaluate the class boundaries. This applies in a similar way to the reverse case, namely, highly imbalanced classes where the target class makes up a small fraction of the training sample. If the class balance is unknown or if the target concept is open, the entire training sample becomes questionable.

When unseen texts are to be classified, standard classification approaches often make unreliable decisions in non-standard tasks because of two difficulties: (1) the classifier has no support, that is, training examples with high similarities to the unseen texts are not available, and therefore its confidence estimation fails, and (2) a different concept might have been learned.

This dissertation is endeavoring to address these problems in non-standard text classification, which are rooted in the text representation and the training sample. Sophisticated models are proposed that capture the gist of non-standard tasks and that have the capability to generalize. In summary, bag of words holds the risk of deluding the learning algorithm for particular tasks, moreover, it is difficult diagnosing this risk with standard evaluation methods, such as leave-n-out validation. Therefore, appropriate experimental evaluations are proposed. Furthermore, difficulties that result from the training samples are approached by proposing strategies for feature engineering (the creation of text representations), sampling (the creation of training samples), learning (the creation of text classifiers), and evaluating text classification solutions.

1.1 Thesis structure

Text classification deals (1) with the discovery of knowledge in texts and (2) with the application of this knowledge for extracting, filtering, or retrieving information in text streams or collections.

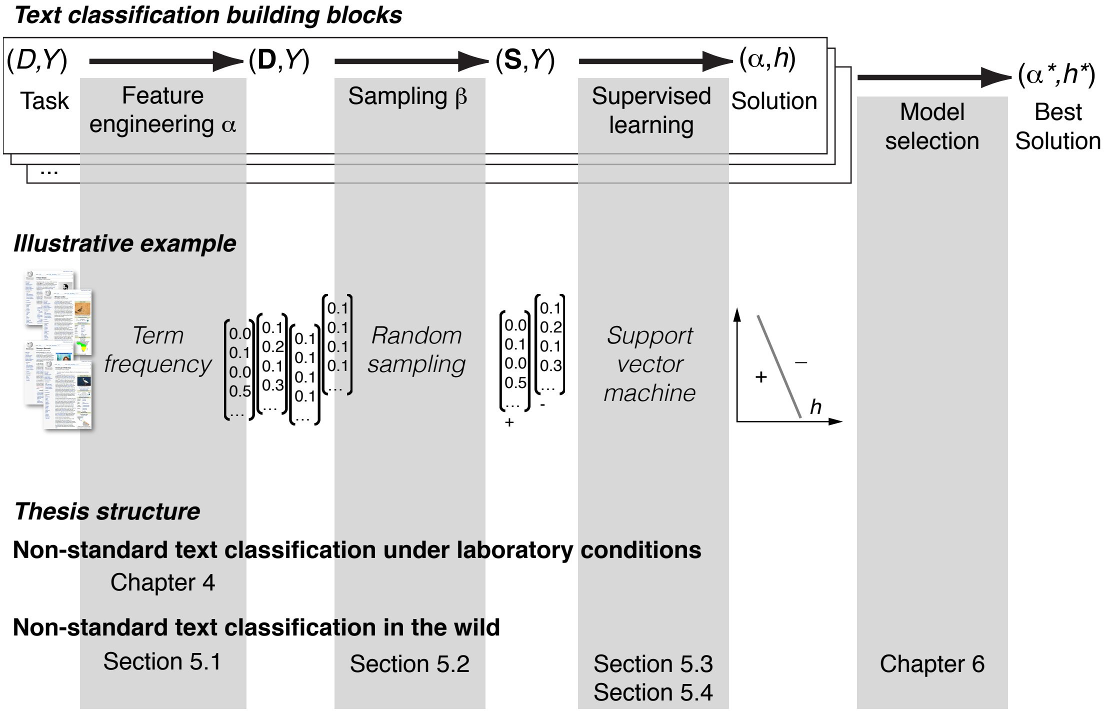


Figure 1.2: *The figure shows the building blocks for text classification (feature engineering, sampling, supervised learning, model selection), which also represent the structure of this thesis. As illustrative example, consider the text classification task (D, Y) that comprises the population D of Wikipedia articles and the class scheme Y with the classes “+, featured articles" and “-, non-featured articles". The articles are represented under a bag-of-words model and a training sample is randomly drawn from Wikipedia. A support vector machine is trained on the labeled training sample.*

(1) The kind and the source of knowledge to be discovered are encoded by a text classification task. A text classification task T, $T = (D, Y)$, comprises a population D, the source of knowledge, and a class scheme Y, the kind of knowledge. The population comprises the texts that are or will be considered during classification. Often, only little information exist concerning D. The class scheme specifies a partition concept, regarding topics, authors, genres, and the like. The so-called target concept c is a generally unknown function that correctly maps the population to the class scheme.

Discovering knowledge is implemented by finding a hypothesis (or classifier) h that is a close approximation of the target concept. A learning algorithm L, which is also known as classifier, induces h from a sample of examples, for which the class membership is known. This induction is known as supervised learning or simply training. For learning from examples, two further functions are needed: a feature engineering function α, which maps a text to the input format of L and a sampling function β, which implements a strategy for selecting examples from D.

(2) The application of knowledge, in terms of classifying texts, is implemented by a text classification solution of T. A solution $m = (\alpha, h)$ comprises a feature

engineering function α and a classifier h. A text is classified by m as follows: α is employed for representing the text and h is employed for predicting its class value.[1]

The process of modeling T is on the "how" of the discovery and the application of knowledge. Modeling is regarded as the process of specifying a model $M = (\alpha, \beta, L)$ that is used for building automatically a solution for T. Informally, modeling is on exploring how examples are represented, how they are selected for training, and how one learns a classifier from them. This is an ambiguous undertaking. The engineer has to gain an understanding of the task at the level of a domain expert along with a profound understanding of machine learning since, most important, the feature engineering α and sampling β need to support the learning algorithm L. Furthermore, the engineer designs and chooses appropriate experiments and measures so that model selection can be applied to find effective solutions for T. Modeling starts either in a top down or bottom up fashion and continues iteratively. Top down: understand the task and then conduct experiments. Bottom up: conduct experiments and then gain a better understanding of the task. The described processes are demonstrated for various text classification tasks in this thesis.

The modeling process for text classification tasks can be summarized in four building blocks, cf. Figure 1.2:

(1) feature engineering
(2) sampling
(3) supervised learning
(4) model selection

In the first three building blocks, engineers define the methods used for building a classification solution; in the fourth block, they choose a strategy for measuring the effectiveness of the resulting solution.

(1) Feature engineering is the process of developing, selecting, extracting, and generating features that represent a text in a metric space. Feature engineering defines the construction of a text representation; more formally, it defines a function α,

$$\alpha : d \mapsto \mathbf{d},$$

where d is a text document represented by a feature vector $\mathbf{d}$. During this thesis d is always, unless otherwise mentioned, an *unstructured* text, a text without metadata such as formatting, revision history, or source information.

(2) Sampling is the process of choosing examples, here, from a stream or collection of texts. Formally, it is a function that selects elements from a population $\mathbf{D}$ of represented documents to form a training sample $\mathbf{S}$ of documents:

$$\beta : \mathbf{D} \mapsto \mathbf{S},\ \mathbf{S} \subseteq \mathbf{D}.$$

[1] Note, the output comes often along with a confidence value or a class probability, which might be used for additional processing, for example, ranking in a retrieval application.

The sample **S** is utilized for learning and evaluating a text classifier. Depending on the text classification task, the examples are labeled with respect to a classification scheme. Since labeling usually entails costs, the size $|\mathbf{S}|$ is restricted to a budget of time, money, or whatever resources are needed.

(3) Supervised learning is the process of learning from examples. A learning algorithm L induces a hypothesis h from labeled examples **S**,

$$L : \mathbf{S} \mapsto h, \; h : \mathbf{d} \mapsto y,$$

where, during this thesis, the class label $y \in \{+1, -1\}$ is binary, unless otherwise mentioned.

(4) Model selection is the process of estimating the effectiveness $e(m, \mathbf{S})$ of a classification solution $m = (\alpha, h)$, given a sample **S**, an effectiveness measure, and an experiment, and selecting the best model accordingly.

The structure of this thesis corresponds to the building blocks in Figure 1.2. Chapter 4 describes analyzing non-standard tasks and feature engineering under laboratory condition, that is, the analyses are based on text corpora with a closed set of classes and examples. The challenge is to understand, from an engineering perspective, the particular classification task and to design a text representation that supports the employed classification technology. Chapter 5 provides research towards text classification in the wild by proposing robust feature engineering, sampling, and learning algorithms. Finally, Chapter 6 contributes to model selection in the wild.

1.2 Contributions

Chapter 2 This chapter provides the background of the most relevant text classification tasks. It provides a categorization scheme for text classification tasks, which is based on the Lasswell Formula, a language function model used in communication theory.

Chapter 3 This chapter establishes a taxonomy of different types of bias found in the literature; a classifier's inductive bias is an explanatory model of the classifier's capability for predicting the class of unseen examples. The chapter also reviews the inductive biases of common supervised learning algorithms, and, finally, gives an introduction to support vector machines, which are often employed in text classification tasks.

Chapter 4 This chapter comprises sections, which examine a variety of text classification tasks and new text representations. In this regard, each section also focuses on tailored experiments and evaluations for the particular task.

Section 4.1 We combine the output of a stemming algorithm (stems) and the stem-reduced words (co-stems) in text representation for non-standard text classification tasks. Besides the content, the writing style is an important discriminator for many tasks. Ideally, the solution of such a task employs a text representation that models both kinds of characteristics. Word stems are clearly content capturing, whereas word suffixes qualify as writing-style indicators. We analyze the discriminative power of our new representation for a broad range of tasks and provide insights into the adequacy and task-specificity of text representation models. For several relevant tasks, co-stem-based representations outperform bag-of-words models.

Section 4.2 We contribute to core-vocabularies for Web genre analysis and discuss existing and new technologies for their construction. So-called core-vocabularies focus on the words that are descriptive for a specific class. Combining concentration measures with core-vocabularies results in non-linear features with a high generalization capability. We also present new evaluation measures and show in a quantitative analysis that our features are superior to state-of-the-art Web genre representations.

Section 4.3 Wikipedia provides an information quality assessment model with criteria for human peer reviewers for identifying featured articles; for the task "Is an article featured or not?" we exploit the articles' distributions of character trigrams. This representation does not require meta-features such as the edit history and aims to evaluate writing style. We conceptualize an experiment design where, among others, the domain transferability is analyzed. Character trigram representations outperform existing methods for this information quality task.

Section 4.4 It is known that language identification can be accomplished with a high precision for ordinary texts; we extend these findings and compare for the first time the effectivenesses of state-of-the-art language-identification approaches on very short, query-style texts. In a multi-language information retrieval setting, the knowledge about the language of a query is necessary for further processing. The results show that already for single words an identification accuracy of more than 80% can be achieved; for slightly longer texts we report accuracies close to 100%.

Section 4.5 We study the applicability of Koppel and Schler's unmasking approach [153] for authorship verification analyses. In these analyses one is given writing examples from an author A, and one is asked to determine whether unseen texts are also written by A. Therefore, unmasking assesses the usage of function words in A's writing examples and unseen texts. The main research question is to what extend unmasking is applicable if the prior writing examples are noisy, that is, not all of them are originally from A. This question is relevant for the subsequent section on intrinsic plagiarism analysis.

Section 4.6 We examine the question whether plagiarized text sections can be detected within an automatized analysis even if no reference is provided, for example, if the plagiarized sections are from a book that is not available in digital form. We refer to this automation as intrinsic plagiarism analysis, which can be transformed into an authorship verification analysis. Therefore, stylometry and, for the first time, one-class classification are utilized for constructing a set of reliable and a set of suspicious writing examples; both sets serve as the input of an authorship verification analysis. We study the effectivenesses of this transformation process and of the entire intrinsic plagiarism analysis.

Chapter 5 This chapter addresses the problems of applying text classification in the wild. The major problems relate to the training sample: first, the labeling of training examples usually involves a great deal of expense and is limited or even impossible; second, the laboratory conditions presumed in almost all classification studies and text corpora do not meet the challenges of text classification problems in the wild, i.e., in the real-world.

Section 5.1 Text classification tasks in the Web deal with collections of enormous size, which makes the ratio between the training sample and the set of unseen texts extremely small. With a sample ratio close to zero, the evaluation of the generalization capability of a classification solution with leave-n-out-methods becomes unreliable and leads one astray. In order to alleviate this problem we introduce the idea of robust models where the engineer intentionally restricts the hypothesis structure within the feature engineering process.

Section 5.2 We explore a sampling strategy for active learning, a subdiscipline of supervised learning, for achieving a greater classification effectiveness with fewer training examples. This is largely accomplished by allowing the active learning heuristics to select the examples from which it learns. Our new, more general sampling strategy is based upon machine learning and learns the informativeness of examples from prior classification tasks.

Section 5.3 We suggest to conceive a text classification task as a one-class classification problem if the complementary class is unrepresentable, for example, because of noise or volatility. As an illustration of this situation, we study the task "Is a Wikipedia article flawed by a flaw f?" since Wikipedia provides cleanup templates for tagging articles that are flawed by f. Untagged articles cannot serve as complementary examples since they might be flawed but not tagged yet, and, more important, since their distribution is dynamic. Tackling this situational condition, we employ a one-class classifier and thoroughly analyze its effectiveness.

Section 5.4 For learning in a one-class classification problem multimodal distributed target classes, we propose a new cluster-based ensemble that outperforms standard one-class technology for several text classification tasks. Various relevant tasks in information retrieval, filtering, and extraction are one-class classification problems at heart; that is, common discrimination-based classification approaches are not applicable. Achieving a high effectiveness when solving one-class problems is difficult and it becomes even more challenging when the target class is multimodal, which is often the case. Our idea is to learn for each mode a separate one-class classifier.

Chapter 6 This chapter covers model selection for text classification solutions when the underlying data source undergoes unknown subclass distribution shifts. This information volatility is observed, for examples, in online media, such as tweets, blogs, or news articles, where the text emissions follow topic popularities. We propose the notion and the estimation of the “expected effectiveness” of text classification solutions under subclass distribution shifts and introduce a probabilistic effectiveness bound for selecting solutions with a superior stability.

Related publications by the author

Used	Publisher	Length	Venue	Type	Year	Reference
in			*Publication*			
4.1	Springer-Verlag	short	ECIR	conference	2011	[178]
	N. Lipka and B. Stein. Classifying with co-stems: A new representation for information filtering.					
4.2	Springer-Verlag	full	–	book chapter	2011	[295]
	B. Stein, S. Meyer zu Eissen, and N. Lipka. Web genre analysis: Use cases, retrieval models, and implementation issues.					
4.3	ACM	poster	WWW	conference	2010	[177]
	N. Lipka and B. Stein. Identifying featured articles in Wikipedia: Writing style matters.					
4.4	Springer-Verlag	poster	ECIR	conference	2010	[99]
	T. Gottron and N. Lipka. A comparison of language identification approaches on short, query-style texts.					
4.5	IEEE	full	TIR	workshop	2008	[293]
	B. Stein, N. Lipka, and S. Meyer zu Eißen. Meta-analysis within authorship verification.					
4.6	Springer-Verlag	full	–	journal	2011	[294]
	B. Stein, N. Lipka, and P. Prettenhofer. Intrinsic plagiarism analysis.					
5.1	IEEE	full	TIR	workshop	2011	[179]
	N. Lipka and B. Stein. Robust models in information retrieval.					
5.2	online	full	DMEF	conference	2011	[270]
	J. Shanahan, N. Lipka, and D. Van den Poel. Learning to active learn with applications in the online advertising field of look-alike modeling.					
5.3	ACM	full	SIGIR	conference	2012	[9]
	M. Anderka, B. Stein, and N. Lipka. Predicting quality flaws in user-generated content: The case of Wikipedia.					
	ACM	poster	CIKM	conference	2011	[8]
	M. Anderka, B. Stein, and N. Lipka. Detection of text quality flaws as a one-class classification problem.					
	ACM	poster	WWW	conference	2011	[7]
	M. Anderka, B. Stein, and N. Lipka. Towards automatic quality assurance in Wikipedia.					
5.4	ACM	poster	SIGIR	conference	2012	[180]
	N. Lipka, B. Stein, and M. Anderka. Cluster-based one-class ensemble for classification problems in information retrieval.					
6	IEEE	short	ICDM	conference	2012	[176]
	N. Lipka, B. Stein, and J. Shanahan. Estimating the expected effectiveness of text classification Solutions under subclass distribution shifts.					
–	ACM	full	CLEF	conference	2009	[6]
	M. Anderka, N. Lipka, and B. Stein. Evaluating cross-language explicit semantic analysis and cross querying.					
–	IEEE	full	CISCHED	conference	2009	[16]
	M. Aufenanger, N. Lipka, B. Klopper, and W. Dangelmaier. A knowledge-based Giffler-Thompson heuristic for rescheduling job-shops.					

Chapter 2

The history of text classification

This chapter provides a landscape of the most relevant text classification tasks. The tasks are organized by a function scheme that is usually used in communication studies. Furthermore, the descriptions of the tasks offer insights into the development history of text classification solutions. The landscape establishes the relationship between standard and non-standard text classification in our conducted research. For a general overview of text classification, many surveys on text classification exist [3, 266, 343, 207].

In communication studies, several well-known models exist which decompose the functions of a text. A *text* is an utterance that can be represented in written language. Throughout this thesis, a document is a written and unformatted text, which is also known as plain or unstructured text in the literature. Table 2.1 summarizes four models of language functions arranged by communication aspects: the four-sides model by Schulz von Thun [264], the Jakobson functions of language by Jakobson [131], the Lasswell formula by Lasswell [163], and the Organon model by Bühler [46]. The Lasswell formula "who says what to whom in which channel with what effect" is often utilized for organizing the research in communication studies and serves here as the basis for organizing text classification tasks.

Text classification is the generic task of making a functional statement about a text. One of the first text classification tasks in history, namely topic categorization, is the determination of what a text is about. Within this task, a classifier makes a functional statement about the subject of a text, the "what" of Lasswell's language function model. In the early 1960's, a lot of research was carried out into automatically assigning a text to one or more given topics. This is driven by text representations derived from the words in the texts, starting with index terms in the papers by Maron and Kuhns [195], Maron [196], and Borko and Bernick [32, 33]. Today, text representations for topic categorization include more sophisticated meta-features that, for example, rely on external knowledge retrieved from Wikipedia [92]. In what follows, we will discover that more and more tasks are related to language functions other than just "what a text is about"; these tasks typically belong to the group of non-standard text classification tasks and need specialized features.

Table 2.1: *Models of language functions.*

Aspect	**Bühler** [46]	**Lasswell** [163]	**Jakobson** [131]	**Thun** [264]
Source	Expressive Function	Who says	Emotive Function	Self-Revealing Layer
Subject	Referential Function	what	Referential Function	Matter Layer
Receiver	Conative Function	to whom	Conative Function	Conative Layer
Medium	–	in which channel	Phatic Function	Relationship Layer
Message	–	with what effect?	Aesthetic Function	–
Code	–	–	Metalingual Function	–

2.1 Who is speaking

Authorship attribution, verification, and profiling are the most prevalent text classification tasks related to the source of a text. The difference between the first two tasks is that within authorship attribution a set of candidate authors with writing examples is given, which is not necessarily the case with authorship verification. The third task, authorship profiling, is on characterizing an anonymous author by dimensions such as gender or age. The text representations, utilized for analyzing these tasks, are often writing-style-related. It should be noted that the writing style of an author depends on the time lapse between text formations, disclosure requirements, and the like, and is not always author-invariant.

2.1.1 Authorship attribution

During the 19th century, the Shakespeare authorship debate arose, questioning whether William Shakespeare's poems and plays were written by himself. One of the suspicious candidates was Sir Francis Bacon. Mendenhall [200] conducted in 1901 possibly the first quantitative analyses of writing style in this context, which compares the word-length distributions observed in Shakespeare's plays and Sir Francis Bacon's texts. Therefore, Mendenhall [199] applied his research on languages and writing style, and his finding was that Bacon could be excluded from the candidates as the stylistic differences are too large. In 1975, Williams [334] rejected Mendenhall's conclusion by showing that the comparison of Shakespeare's verse and Bacon's prose is inappropriate. An author is likely to produce different writing styles when writing verse and prose, therefore one cannot draw conclusions related to the authorship from these differences. On the contrary, same writing style is a good indicator for same authorship, and, however, the Shakespeare authorship question was the first prolific authorship attribution task tackled by a statistical analysis.

A second famous case is the disputed authorship of the Federalist Papers, which is one of American history's most infamous questions. Between 1787 and 1788, 85 articles were anonymously published in several of New York's newspapers with the goal to promote the ratification of the United States Constitution. While the authorship of the articles had been largely clarified, twelve of them were disputed. In 1964, Mosteller and Wallace presented a reliable authorship analysis based on the frequencies of a set of function words (“the”, “and”, “for”, etc.). The usage of function words is still today a strong feature within authorship tasks.

Within the field of authorship attribution various features evolved. In 1939, Yule [348] introduced the sentence length as a statistical characteristic for authorship questions. Later, Yule [349] proposed word-frequency distributions, the characteristic K, alphabetical distributions, and vocabulary distributions. More detailed, the word-frequency distribution determines how many words occur once (hapax legomenon), twice (dis legomenon), and so on. Yule’s K is a measure of lexical repetition where $1/K$ is the probability that two randomly chosen words are the same assuming the word occurrences are governed by the Poisson Law. The alphabetical distribution is the frequency distribution of initial letters. Finally, the vocabulary distribution is the distribution of a set of words over a set of classes: given n classes, determine the number of words that occur in $n, n-1, \dots, 1$ classes. Additionally, Yule analyzed each part of speech (noun, verbs, adjectives, pronouns, etc.) separately. His work laid the foundations of authorship analysis features.

In the 1960’s and 1970’s more lexical-complexity measures such as Yule’s K were developed. Among them are Herdan’s V_m, Sichel’s S, Brunet’s W, Honoré’s R. All of them are based on word frequencies and the size of the used vocabulary. Furthermore, a good deal of readability measures, such as the Automated Readability Index, LIX, SMOG Grading, or Flesch-Kincaid readability test, were published in these decades. “Non-traditional authorship attribution” was born, and stylometry, which is the field on quantifying writing style, makes up the largest part of it. Informally, the goal of stylometry is to identify text features (style markers) that are writer-invariant; features that only tend to similar values or characteristics when the represented texts are from the same author—often only the combination of features become writer invariant.

2.1.2 Authorship verification and intrinsic plagiarism analysis

Authorship verification first occurred in the 21st century with the objective of verifying a singular authorship of a set of texts. In contrast, intrinsic plagiarism analysis works on one text as input, but the goal basically remains the same. Trivially, the input text has to be chunked in a set of text sections for reducing an intrinsic plagiarism task to an authorship verification task. Even though the applied features are the same as in the field of authorship attribution, the statistical evaluation is fundamentally different. Authorship verification is a one-class classification problem because no reliable examples from the complementary class are available. Without the complementary class, which represents all foreign authors, it is not possible to determine the discriminativeness of features. van Halteren [314] reformulated the task as a two-class problem by representing the complementary class by a large reference corpus. Koppel and

Schler [153], Koppel et al. [155] worked with the reformulation "are two texts written by the same author" and suggested an analysis of the learning curves that are formed by impairing the text representation. In the context of this thesis, machine-learning-based one-class classification technology [303] has been employed for the first time in intrinsic plagiarism analysis, cf. Stein, Lipka, and Prettenhofer [294].

2.1.3 Authorship profiling

Authorship profiling, which is closely related to sociolinguistics, encompasses all tasks that determine further information about the author. The field of sociolinguistics arose in the beginning of the 20th century and concerns the relationship between society and language. Sociolinguistic analyses provide information of the social contexts of authors. The important subfield of dialectology arose slightly earlier, in the 19th century, and concentrates on the variety of language, which is a so-called "lect". The variety of language over individuals is known as "idiolect", which is the linguistic perspective on authorship attribution and verification. The variety is known as "sociolect" if it depends on social classes, and it is known as "regiolect" if it depends on geographical areas. Lects can also depend on historical eras (stylochronometry) or on political groups, which can give clues to the intention of a text. Even though dialectology was around for a long time, the automatic classification of dialects was studied much later, for example in 2006 by Huang [126]. More examples of state-of-the-art authorship profiling are on classifying the nativeness of language use [315], the author's gender [120, 262], the author's age [262, 227], or the author's personality [228, 14, 205].

2.1.4 Stylochronometry analysis

Stylochronometry analysis is the task of determining the creation date of a document, which is not only relevant to historians but, for example, to all Web search users, who like to chronologically filter retrieved documents. For analyzing documents without meta-information, either writing-style or content-based methods are employed. One of the first works determining the creation date via language models based on word unigrams was by de Jong et al. [68] in 2005. This approach was improved by Kanhabua and Nørvåg [138] with advanced features such as part-of-speech tags and collocations. Other studies examined a variety of content-based features: Swan and Jensen [300] studied the temporal dimension of word usage, and, on the other hand, Garcia-Fernandez et al. [95] considered dates of entities occurring in documents extracted from Wikipedia and also neologisms and archaisms extracted from Google books *n*-grams.

2.1.5 Provenance and lineage analysis

Provenance and lineage analyses unveil information about a document's origin and history of ownership. Documents in the Web are copied, modified, and stored in different places. Knowing the origin of documents can help, for example, to judge its trustworthiness. While this field is well studied for scientific data processing and database management systems [275, 34, 56], to our knowledge, the automatic identification of provenance for text documents in general has not been studied by now.

2.2 What is the text about

The classification of texts with respect to their *topic* (or subject) dates back to the first libraries around 2600 BC. The libraries in Nippur about 1900 BC had the first library classification systems. In general, a library classification system categorizes documents by index terms that describe what a text is about.

In the 1920s and 1930s the first automatized information retrieval systems were developed. In 1931, Emanuel Goldberg patents a document search engine, which uses pattern recognition techniques for retrieving microfilmed documents based on their metadata. Hans Peter Luhn started in 1947 developing an information retrieval system for chemical compounds based on punch cards. And, in 1950, Mooers [209, 210] used the term *information retrieval* for the first time.

Text categorization relies on indexing technology used in information retrieval. The entry of probability theory in information retrieval has it's own history, starting with Maron and Kuhns [195] in the 1960's, followed by Cooper, Robertson [249], and van Rijsbergen [316, 317]. Probabilistic topic categorization was operationalized for the first time by Maron [196] in 1961. Even though automatic information retrieval systems focus on the idea of ranking documents by their relevance to an information need, which is formulated as a query, ranking and categorization are conceptually the same. The two most influential projects in information retrieval and topic categorization are from Salton and Lesk [254] and Fuhr et al. [91]. Salton and Lesk founded in the 1960's the SMART information retrieval system, where the vector space model was invented and implemented, along with statistical term weighting schemes [255]. Later, the vector space model was extended to the generalized vector space models [337] and the topic-based vector space models [18]. In contrast to the SMART project, Fuhr et al. [91] founded the AIR/X system, which is one of the first automatic text categorization systems that handles numerous categories and documents. Then, in the 1990's, topic categorization became more appealing and found application in many filtering, organization, and dispatching tasks, as well as in the generation of metadata [266].

2.3 To whom is the text addressed

In general, texts have implicit or explicit audiences, where an audience can be a group that speaks the same language, private persons, adults, or maybe children. An implicit audience is a group of people who can read and understand a text whereby we identify language identification as a task related to implicit audiences. An explicit audience is a group of direct addressees whereby adult and private content detection are related to explicit audiences.

2.3.1 Adult content detection

The protection of children is a matter of concern to all of us. The classification of adult content is becoming ever more important since the Web offers pornographic material in masses.[1] For spotting this content, Santos et al. [259] employed compression procedures that also had been shown to be effective in spam classification by Bratko et al. [38], and, moreover, Kim [146] classified texts in a four level grading scheme via standard classifiers. Although, this task is relatively new and arose with the commercialization and popularization of the Web, more related work can be found in Ho and Watters [117] and Hu et al. [124].

2.3.2 Private content detection

A private text has specific recipients and several types of information, such as security information, corporate trade secrets, or personal content, should not be read by others. In contrast, a public text is allowed to be broadcasted. Nguyen [216] applied statistical learning methods to the task of discriminating between private and public texts. The analyzed datasets comprise emails from the Enron Email dataset serving as private texts, versus Twitter messages, Myspace forum discussions, and Slashdot comments serving as public texts. The Enron Email dataset contains emails of the U.S. energy company Enron Corporation, which are accessible because of the investigations in one of the largest cases of accounting fraud in 2001. In a 1 : 1.12 setup (public : private), 95.6% accuracy in discriminating the data can be achieved.

A highly related task is “data loss detection/prevention”. Its goal is to detect texts that are private but readable outside the authorized scope of recipients and to prevent unauthorized readability. Usually this task refers to data in general, which can be in-use, in-motion, and at-rest [269, 225]. With the growth of cloud computing the demand for data loss detection systems is likely to be increased in the private and in the commercial sector. The aspect of identifying

[1] At the upcoming WSDM 2013 will be a dedicated workshop on this topic, called SEXI 2013 (Workshop on Search and Exploration of X-Rated Information).

data, for example, credit-card data, intellectual properties, or social security numbers, range from precise (via hashing) to imprecise (via regular expressions and statistical machine learning) identification approaches.

2.3.3 Language identification

Language identification deals with the identification of a text's language. This task is seen as solved for conventional texts, and current research is on language identification for special use cases, such as short texts or queries, for example, by Gottron and Lipka [99]. Ingle [130] is one of the first who proposed a statistical approach for identifying the language of a text. In the 1990's, Dunning [77] and Cavnar and Trenkle [51] studied n-gram models as a representation for the classification task. Standard approaches are summarized in the comparative study by Lena Grothe and Nürnberger [168].

2.4 Which channel is used

The channel that is used for publishing a text is often an implicit outcome of text classification tasks. We propose Web genre classification with appropriate schemes, for example, the differentiation between Wiki articles, blog posts, forum entries, private or commercial homepages, for identifying channels. Web genre classification is, however, a special case and is examined on its own merits in Section 2.6.

Also the grading of text with respect to readability [162, 64, 86, 147, 52] can be seen as a channel-related task. The popularization of these grading schemes and indexes is driven by the military to minimize the risk of misunderstandings [142]. It should be noted that the readability of a text is also audience-related.

2.5 What is the effect of the text

Modern research in text classification is on the *effects* of texts. The questions behind the respective tasks are, for example: is a text an act of vandalism; is it making people smile; or does it express a sentiment. Especially the last task received much attention recently.

2.5.1 Opinion mining

A large part of research in opinion mining concerns movie and product reviews, see [223] for a detailed survey. In the context of product reviews, additional information extraction is needed to identify product features and consecutive sentences [182, 123, 231]. Building a list of product features is a classification task by itself, known as terminology extraction [137, 35]. It is worth mentioning the subtask of retrieving comparative sentences, where two or more entities are compared with respect to a product feature [94]. In contrast, opinion mining for movies is more straightforward and requires less preprocessing.

The central elements in mining opinions are sentiment, polarity, and subjectivity analyses. Early related work was already conducted in the 1970's by Carbonell [48] and later by Wilks and Bien [333]. This was followed by research in identifying the point of view in texts [332, 331, 110, 253]. Later, Hatzivassiloglou and McKeown [109] and Wiebe [330] deal with the semantic orientation of adjectives, before sentiment analysis became a vibrant research field in 2001 [65, 302, 307, 72, 212, 66, 222], followed by hundreds of publications. Different learning setting and specialized features were examined: unsupervised approaches [223], domain transfer [185], polarized word lists enhanced by synonyms, antonyms, negations, or emoticons [145, 65, 243].

Sentiment analysis is of high importance for economics, and therefore detecting opinion spam is relevant [132]. A distinctive example for sentiment analysis in economics is the work by Archak et al. [12]; it studies how important specific product features are, and, regarding these features, how the polarity of a review affects the customers' buying decisions. This is only one demonstration of the impact of opinion mining. Today's big data experts consider sentiment analysis as a key technology for retrieving valuable information for making strategic decisions in companies.

2.5.2 Vandalism detection

Vandalism is a phenomenon that occurs in editable online content such as Wikipedia articles. For the case of Wikipedia, "vandalism is any addition, removal, or change of content in a deliberate attempt to compromise the integrity of Wikipedia." [2] Depending on the vandalized domain, the types of features can go beyond capturing textual characteristics and evaluate editor profiles, edit histories, or other meta-information.

In a study by Potthast et al. [235], vandalized and well-intentioned Wikipedia edits are discriminated by a logistic regression classifier using features such as the longest consecutive sequence of the same character, the compression rate, the frequency of upper case letters, and the frequency of vulgar words. Meta-information is utilized as well, such as the anonymity of the editor, the length of supplied comments, and the similarity between the old and updated articles.

The range of features was expanded by Wang and McKeown [326] where punctuation misuse (e.g. "!!!"), Web slang (e.g. "LOL"), comment cue words

[2] http://en.wikipedia.org/wiki/Wikipedia:Vandalism accessed 12-September-2012

were suggested. Chin et al. [57] utilized the revision history to build language models for vandalism detection in Wikipedia. In addition, Wang and McKeown [326] considered the probabilities that edits are generated by syntactical and semantic *n*-gram language models function as features.

Based on the hypothesis "vandalism can be detected by writing style", Harpalani et al. [106] proposed probabilistic context free grammars (PCFG) for unveiling syntactic patterns. Basically two PCFG parsers are trained, one on vandalized and one on well-intentioned Wikipedia edits. A parser ouputs a probability of generating an edit, and, for the classification of an unseen edit, the corresponding probabilities of both grammars are compared. Other features in this context relate to an author's reputation [1] and the impact of an edit [237].

2.5.3 Other tasks

There are several uncommon tasks that are related to the effect of a text. Koppel et al. [156] studied ideology identification, which is, for example, the discrimination between speeches of Republicans or Democrats. Yu [347] evaluated text classification methods for classifying erotic language patterns in poems (eroticism) and for classifying sentimentality levels in texts (sentimentalism). Yang et al. [340] classified emotions expressed in blog entries; Mihalcea and Strapparava [204] classified humor.

Carvalho et al. [50] and Reyes et al. [246] detected irony in user posts of newspaper and tweets, whereas Gibbs [96] provided a descriptive analysis of irony types (hyperbole, sarcasm, rhetorical questions, understatements, and jocularity) and Filatova [83] crowdsourced a corpus for irony and sarcasm classification in Amazon reviews.

Further common tasks are the classification of ads, fake reviews, patents, frauds, and lies. In natural language processing, there are even more tasks such as part-of-speech tagging, grouping adjacent words, and anaphora resolution. In field of information extraction the list continues with named entity recognition, named entity detection, relationship extraction, and comment extraction. And, the number of text classification tasks is still growing.

2.6 A comprehensive task: Web genre analysis

Web genre analysis is of high practical interest and provides information, with respect to nearly all language functions, cf. Table 2.1, depending on the assigned class scheme. Web genres relate to the presentation, the intended target group, the effects, the channel in use, and the authorship of a text. Regarding the Web, due to different user groups and technical means, several favorite specializations of Web documents emerged: a document may contain many links (e.g. a link collection), scientific text (e.g. a research article), almost no text but pictures (e.g. an advertisement page), or a short answer to a specific question (e.g. a message in a help forum). These are examples for, what is called here, “genre” or “Web genre”.

This section outlines the use cases in [295], where Web genre analysis forms an essential building block in the information processing chain, namely retrieval services that are empowered by genre labeling and information extraction that uses genre classification as auxiliary technology. Web genre analysis demonstrates two general observations for text classification tasks:

- The use cases of text classification tasks can be manifold.
- Tasks can be related to more than one language function.

2.6.1 Genre-enabled Web search

Search engines are the most influential and important applications in the Web. An integration of genre analysis can happen according to two different paradigms, namely filtering and Web search. Under the filtering paradigm, a user declare their information needs in terms of genre preferences, and the retrieval process accounts for these constraints. Under the classical Web search paradigm using Google, Bing, Yahoo, and the like, Web genre information is introduced by assigning genre labels to the snippets in the search results. Both approaches have their advantages and disadvantages, pertaining to retrieval time and retrieval precision. Different Web genre schemes along with technology for identifying the genre classes are compiled in Table 4.5.

2.6.2 Information extraction based on genre information

Web genre schemes provide a diversification of documents into text types that is oriented at search habits and the emerged culture of Web presences. In a technical sense, Web genre models can be understood as a collective term for classification models that quantify arbitrary structure and presentation-related document features, while being topic-orthogonal at the same time. Examples for high-level Web services that need a special text type as input are:

Market forecast summarization Market forecasting seeks to anticipate the future development of new technologies at an early stage. It is vital for most companies in order to develop reasonable business strategies and to make appropriate corporate investments. Market forecasting can be supported by automatically collecting, assessing, and summarizing information from the Web into a comprehensive presentation of the expected market volume. For this purpose a four step approach was implemented by Stein and Busch [289]: collecting candidate documents, report filtering, time and money identification, and phrase analysis along with template filling. The third and fourth steps are computationally very demanding, and the rationale of the proposed approach is reducing unnecessary natural language processing effort by a reliable identification of interesting business reports published on the Web. The heart of this strategy is a genre analysis in the report filtering step.

Retrieval of scholary material Specialized search engines and technology for vertical search are building blocks of future information extraction applications for the retrieval of academic research material. They shall be able to identify, synthesize, and present Web documents related to exercises, FAQs, introductory readings, definitions, or sample solutions, given a topic in question. The driving force is a reliable document type and genre analysis.

Focused crawling for plagiarism analysis The discriminative power of a genre classifier can also be utilized at the crawling stage. Here, the challenges result from a classification model that has to get by with few and small document snippets. An interesting application is plagiarism analysis, which focuses on research articles, book chapters, and theses.

Chapter 3

Supervised learning

Supervised learning means to induce from labeled training examples a function, known as hypothesis or classifier, that predicts the labels of unseen examples. The ultimate goal in statistical classification is to find a hypothesis h that is a close approximation of the target function c, which is generally unknown and therefore the closeness of both functions cannot be directly evaluated. The most fundamental question in supervised learning is: if h can generalize, i.e., predict the classes of unseen examples. The need of these predictions is prevalent in all text classification tasks. This chapter reveals an understanding of supervised learning that forms the foundation of this thesis.

3.1 An explanatory model of generalization: inductive bias

Each hypothesis h, as part of a solution of a classification task T, makes, be it implicitly or explicitly, a-priori assumptions about T. These assumptions will be introduced in what follows as *inductive bias*. The inductive bias forms the rationale for learning—better: for generalization; without bias is no generalization possible [206, 108].

Let h be a hypothesis selected by an inductive learner L when given the examples of a training sample $\mathbf{S}$ to L. An example in the training sample is a pair of an input variable $\mathbf{d}$, which represents the document d, and an output variable y, which represent the class label of the d. Using h, the class label of an unseen example $\mathbf{d}$ can be computed, $h(\mathbf{d})$, precisely stated: "the prediction $h(\mathbf{d})$ follows inductively from $\mathbf{S}$, L, and $\mathbf{d}$", which is the semantics of the formula below:

$$L \wedge \mathbf{S} \wedge \mathbf{d} \ \succ\ h(\mathbf{d})$$

Because of the inductive situation, the predicted class label $h(\mathbf{d})$ need not necessarily correspond to the true class label $c(d)$. As a consequence, $h(\mathbf{d})$ is not provably correct. Mitchell [206] asks what minimum set of additional assertions B could be added to $L \wedge \mathbf{S} \wedge \mathbf{d}$ so that $h(\mathbf{d})$ follows deductively:

$$L \wedge \mathbf{S} \wedge \mathbf{d} \wedge B \ \models\ h(\mathbf{d})$$

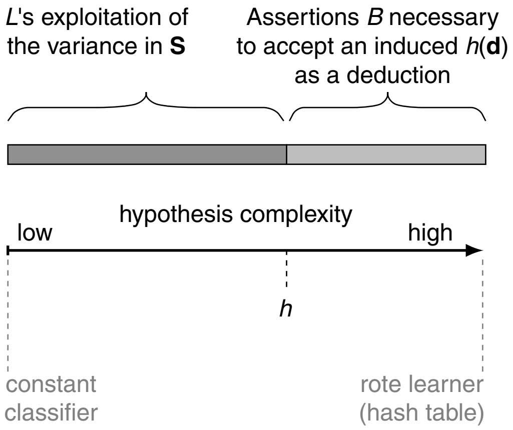

L's exploitation of the variance in S
Assertions B necessary to accept an induced h(d) as a deduction
hypothesis complexity
low
high
h
constant classifier
rote learner (hash table)

Figure 3.1: *Mitchell's inductive bias is defined as the size or complexity of the smallest set of assertions B. For example, a rote learner classifies without any assertions training examples provably correct but is not able to classify unseen examples. This learner has therefore no inductive bias and does not generalize. In contrast, a constant classifier has a strong inductive bias, namely, the assertion $h = c$.*

The inductive bias of a learner L is defined as the smallest set of assertions B such that for all $\mathbf{d} \in \mathbf{D}$ the class label $h(\mathbf{d})$ can be deductively inferred, cf. Figure 3.1. If the additional assertions B are fulfilled, the deduced class label $h(\mathbf{d})$ is equal to the true label $c(d)$; hypothesis and target concept are equal $h = c$.

Even though this understanding of the inductive bias is appealing, this bias is hard to be quantified: typically B represents a formula in predicate logics, specifying the assertions in an axiomatic way, while there is no intuitive and sensible calculus for measuring the complexity, the extent, or the scope of the propositions contained in this formula. Describing the inductive bias of a learner L, either formally or informally, however, describes L's gist. It tells us how much of the variation in a sample **S** is captured by the inductive learner, which in turn correlates with the complexity of the hypothesis space H, which comprises all hypotheses that L can generate.

The idea behind Mitchell's inductive bias becomes perspicuous if two extreme strategies of inductive learners are considered, the here called maximum-bias learner and the zero-bias learner, cf. Figure 3.1. The maximum-bias learner, for example, a constant learner, is able to classify an example **d** without knowing anything about it. Its decision is always the same and nothing of the variance contained in a sample **S** is modeled. By contrast, the zero-bias learner captures the entire variance in **S**, even if its elements are randomly distributed. An example for this extreme is a hash table. Hash tables model arbitrary hypotheses, which comes at the price that from an unseen example **d** nothing can be induced with respect to $c(d)$. Consequently, the generalization ability of a hash table is zero.

Table 3.1: *A taxonomy that relates the different types of bias found in the literature.*

Inductive Bias		
Structural bias $bias(H)$ (learner-independent)		Estimation bias $bias(L)$ (learner-dependent)
Model formation bias	Hypothesis complexity bias	Preference bias

3.1.1 Bias types

The main findings in computational learning theory are based on results in the fields of combinatorics (how many hypotheses are consistent with a given training sample) and probability theory (how likely is the selection of a hypothesis with an error smaller ϵ). Since the hypothesis space H comprises the hypotheses a learner L infers from $2^{\mathbf{D}}$ (all possible training samples), its characteristics take a major role in this context, which are affected by different types of biases.

The types of biases found in the literature are organized within the taxonomy shown in Table 3.1. Commonly accepted is the distinction between a structural bias and an estimation bias, cf. Line 2 in the table. The former is learner-independent and is also known as model bias; the latter is learner-dependent.

Line 3 in the table shows three types of inductive bias that have an impact on the hypothesis space H in terms of its size and elements. The types are: (1) model formation bias, in the form of heuristics and assertions in the feature engineering phase, (2) hypothesis complexity bias, which results from the structure of the hypothesis function, and (3) preference bias, which is a consequence of the optimization strategy of the learner. The feature engineering and the resulting representation restrict H in a structural way and introduce a learner-independent bias. The hypothesis complexity relates to the flexibility of modeling even small differences between randomly drawn samples $\mathbf{S}$. Hastie et al. [108] use the term "model complexity" in this connection. Mitchell [206] subsumes this kind of structural property under the term "restriction bias"; it depends on the structure of a hypothesis h, which is determined by the parameter number and parameter interaction. The exploration strategy defines whether or not H contains all hypotheses of a certain structure. The situation that H is not complete can also be understood as a complete hypothesis space H' that is incompletely explored, such that, the "effective size" of H' corresponds to $|H|$. Mitchell subsumes this kind of structural property under the name "search bias" or "preference bias"; it is determined by the learner.

Finally, all types of bias entail an incurred bias, which cannot be controlled; incurred biases are caused by statistical deficiencies, by shortcomings in design ot the learning algorithm, or by a lack of appropriate features. Furthermore, some of the mentioned structural properties influence each other, while others are orthogonal. Altogether they can be used for reducing $|H|$ and improving the generalization capability of a text classification solution $m = (\alpha, h)$.

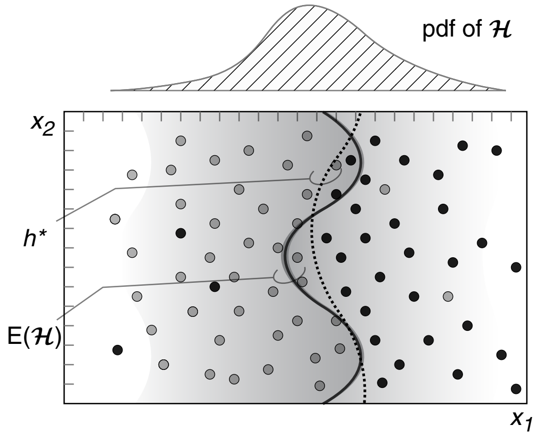


Figure 3.2: *Illustration of the hypothesis space that a learning algorithm produces when learning from different training samples. h^* is the optimum hypothesis and* $\mathrm{E}(\mathcal{H})$ *is the expected hypothesis. In the upper part, the probability density function of $\mathcal{H}$ can be seen.*

3.1.2 Bias quantization

The following definitions unify the notion of biases. The statistical learning theory considers **d** to be a realization of the real-valued multivariate random variable $\mathcal{X}$ and, for simplicity, the assigned class label y to be a value of the binary random variable $\mathcal{Y} \in \{-1, 1\}$ with the joint probability distribution $\mathrm{P}(\mathcal{X}, \mathcal{Y})$. The symbol $\mathcal{H}$ denotes a random variable whose observed values are hypotheses in the hypothesis space H.

Most important is the generalization error of a text classification solution. The minimization of this error is the ultimate goal since it concerns the entire population of the classification task.

Generalization error of a hypothesis h, $err(h)$:

$$\begin{aligned} err(h) &:= \mathrm{E}(loss(h(\mathcal{X}), c(\mathcal{X}))) \\ &= \int loss(h(\mathbf{d}), c(d))\, \mathrm{d}\mathbf{d}, \end{aligned}$$

where

$$loss(y, y') := \begin{cases} 0 & \text{if } y = y' \\ 1 & \text{otherwise.} \end{cases}$$

The generalization error is also known as prediction error, real error, or true error [108, 313, 70, 335, 261]. Usually the population cannot be analyzed in its entirety, and one resorts to an estimator of $err(h)$. The most common estimator of $err(h)$ is the sample error $err_{\mathbf{S}}(h)$, which is defined as follows:

$$err_{\mathbf{S}}(h) := \frac{1}{|\mathbf{S}|} \sum_{\mathbf{d} \in \mathbf{S}} loss(h(\mathbf{d}), c(d)),$$

where $\mathbf{S}$ denotes a sample of examples. $err_{\mathbf{S}}(h)$ is known as "training error" if $\mathbf{S}$ is used for the construction of h by a learner L; it is known as "test error" otherwise.

The generalization error $err(h)$ describes a property of a *single* hypothesis h. In addition, there are also statistics that measure properties of a hypothesis space H such as the structural bias, as well as statistics that measure properties of a learner L such as the estimator bias [90], cf. Table 3.1 and Figure 3.2.

Structural bias of a hypothesis space H, $bias^*(H)$:

$$\begin{aligned} bias^*(H) &:= err(h^*) \\ &:= \min_{h \in H} err(h). \end{aligned}$$

The structural bias quantifies the expected difference between an optimum hypothesis $h^* \in H$ and the target concept c; Hastie et al. [108] terms it model bias. The optimum hypothesis h^* is defined as the hypothesis in H with the lowest generalization error. Depending on the structure of H, h^* does not necessarily need to be equal to c.

Estimation bias of an inductive learner L, $bias(L)$:

$$\begin{aligned} bias(L) &:= \mathrm{E}\left(loss(\mathcal{H}(\mathcal{X}), h^*(\mathcal{X}))\right) \\ &= \int loss(\mathrm{E}(\mathcal{H})(\mathbf{d}), h^*(\mathbf{d}))\, \mathrm{d}\mathbf{d}, \end{aligned}$$

with

$$\mathrm{E}(\mathcal{H})(\mathbf{d}) = \underset{y \in Y}{\operatorname{argmax}} \int \mathbf{1}(h(\mathbf{d}), y)\, \mathrm{d}h$$

and

$$\mathbf{1}(y, y') := \begin{cases} 1 & \text{if } y = y' \\ 0 & \text{otherwise.} \end{cases}$$

The distribution over the hypotheses in H is characterized by the random variable $\mathcal{H}$, which can be estimated by learning from randomly drawn training samples. The estimation bias quantifies the expected difference between the expected hypothesis returned by L and an optimum hypothesis $h^* \in H$ [206, 108]. The expected hypothesis is not necessarily an element of H. Informally speaking, the expected hypothesis $\mathrm{E}(\mathcal{H})$ classifies the document $\mathbf{d}$ as follows: $\mathbf{d}$ receives the class label that is predicted by the majority of hypotheses in H, cf. $\mathrm{E}(\mathcal{H})(\mathbf{d})$.

The structural bias is independent of the distribution of hypotheses and considers only the structural aspects of the hypothesis space. In contrast, the estimation bias is based on the distribution and is therefore learner-dependent. Both characteristics are practically not computable since it is not possible to identify the optimum hypothesis h^* without exhaustively exploring the complete hypothesis space. The expected hypothesis can be estimated by repetitive resampling and learning. When approximating h^* by the target function c, which is known for the training examples, the estimation bias can also be estimated. The quality of this estimate is unknown but, nevertheless, the estimate can be applied for comparing different text classification models.

3.2 Inductive biases of common learning algorithms

The inductive bias of a learning algorithm describes its way of learning from examples. Understanding the inductive bias of learning algorithms is an essential part both in applying and in developing them. It also enables the engineer to explain the effectiveness and generalization capability of an algorithm to a large extent. This section lists the most common learning algorithms used in text classification and describes their inductive biases.

Linear support vector machine (SVM) An SVM is a maximum margin classifier. The learning algorithm finds a linear classification boundary (hyperplane) that separates the training examples and maximizes the margin to the closest training examples regarding their class labels. A soft margin SVM, proposed by Cortes and Vapnik [59], also allows the misclassification of examples to a certain degree, which is needed when the examples are not separable by the classification boundary. A variety of research studies, for example, by Joachims [134], [133], showed that linear SVMs have good generalization capabilities and rank for most text classification tasks under the top classifiers. In general, a learner with a maximum margin inductive bias selects consistent hypotheses that are maximizing the distance to the closest training examples with different labels.

Naïve Bayes (NB) The naïve Bayes classifier is a probabilistic learning algorithm, which assigns an unseen example to the class that has the highest conditional probability under a parametrized model. Its inductive bias maximizes the conditional independence. The training phase is made up of computing the maximum likelihood estimate of the classifier's parameters. Within the field of text classification the class probability estimates of naïve Bayes are often poorly calibrated but the classification decisions are still reliable [105]. SVMs are often the better choice in general but it was shown that naïve Bayes classifiers are robust to the concept drift problem where the distribution of classes or subclasses changes over time [88].

Decision tree (C4.5) The C4.5 algorithm by Quinlan [239] classifies an unseen example with a decision tree. Each path in a decision tree is a conjunction of constraints on the feature values. The tree is constructed by choosing for each node the feature and constraint which maximizes the information gain in the resulting split of training examples. Often, decision trees have a bias that controls their complexities. For example, a learner with a minimum-description-length inductive bias selects the simplest hypothesis in terms of its description length. This bias is typically implemented by pruning strategies for tree learning algorithms, which prefer smaller trees, for example, in terms of the number of nodes, the maximum path length, or the number of leafs.

k-Nearest Neighbor (k-NN) The k-NN learning algorithm stores the training examples and assigns the most frequent class among the k nearest training examples to an unseen example. Usually the Euclidean or Hamming distance determine the neighborhood of examples. For small k, k-NN is sensitive to the local distribution of the training examples and tends to overfit. The nearest neighbor classifier maximize the distance between the closest examples and has therefore a maximum margin bias. This is not necessarily true for the k-nearest neighbor algorithm with $k > 1$. These learners have a so-called nearest-neighbor inductive bias that assigns an example to the class majority of its neighborhood. This bias is connected to the clustering assumption; that is, similar documents have similar labels.

3.3 Optimization for learning algorithms

Training or learning is the process of selecting a hypothesis based on the available training examples. In many cases the training process is done via optimization, whereas the inductive bias of a learning algorithm is nested in the objective function and the constraints of the optimization problem. The hypothesis that minimizes the objective function is neither necessarily the best hypothesis nor the target concept.

Learning algorithms can be divided into two groups, namely generative models and discriminative models [215]. Generative models estimate the conditional probability density functions. Often no numerical optimization is needed and the hypotheses can be computed in a closed form, which is for example the case for naïve Bayes or linear discriminant analysis. The name linear discriminant analysis might be confusing but the approach belongs to the group of generative models; here, the conditional probability density functions are assumed to be normally distributed, and they are parametrized during the training. Discriminative models are not able to regenerate the observed examples but to discriminate them; they are often trained via optimization. Examples for discriminative models are support vector machines, perceptrons, or logistic regression algorithms.

Linear support vector machines take an important role in text classification and are therefore introduced in this chapter. The feature space is often high-dimensional as it is often spanned by large vocabularies. Linear support vector machines implement an implicit sampling strategy by focusing on examples (support vectors) that are close to the class boundary. The strategy is effective as it avoids the curse of dimensionality and reduces the computational effort of training. Because of the high dimensionality in text classification, a linear class boundary is often sufficient for approximating the target concepts and more complex hypothesis classes are less common. Therefore, we do not go into the details of kernel learning, which is usually part of support vector machine literature since it allows the learning of complex hypotheses. We introduce in the following subsections the basic concepts of optimization for a better understanding of formulating training processes as optimization problems.

3.3.1 Standard form

The standard form of an optimization problem is

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \; i = 1, \dots, m \\ & h_i(x) = 0, \; i = 1, \dots, p, \end{aligned}$$

with the objective function $f(x)$, the inequality constraints $g_i(x) \leq 0$, and the equality constraints $h_i(x) = 0$ and $f, g_i, h_i : \mathbb{R}^n \to \mathbb{R}$, see [36].

A quadratic programming problem is of the form

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) = \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x} \\ \text{subject to} \quad & g_i(x) \leq 0, \; i = 1, \dots, m \\ & h_i(x) = 0, \; i = 1, \dots, p, \end{aligned}$$

where Q is a symmetric quadratic matrix.[1]

An optimization problem is convex, if the objective function f and the inequality constraints $g_1, \dots, g_m$ are convex and the equality constraints $h_1, \dots, h_p$ are affine. The key property of convex optimization problems is that a local minimum is simultaneously the global minimum and corresponds to their optimal solution.

3.3.2 Lagrangian

The Lagrangian of an optimization problem in the standard form is

$$L(x, \lambda, \nu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{i=1}^{p} \nu_i h_i(x),$$

[1] If Q is positive semidefinite, then f is convex, and if Q is zero, then the problem is a linear program.

with the Lagrange multiplier λ_i for the ith inequality constraint and the Lagrange multiplier ν_i for the ith equality constraint. The term $\lambda_i f_i$ is an underestimating approximation of the indicator function $I(f_i(x))$, which makes interpretation of the constraint integration in the Lagrangian more obvious:

$$I(f_i(x)) = \begin{cases} 0 & \text{if } f_i(x) \leq 0 \\ \infty & \text{otherwise.} \end{cases}$$

The Lagrange dual function is

$$g(\lambda, \nu) = \inf_x L(x, \lambda, \nu).$$

The optimum solution has the value p^* and for $\lambda_i \geq 1$, $i = 1, \ldots, m$

$$g(\lambda, \nu) \leq p^*.$$

Therefore the Lagrange dual problem, which is a convex optimization problem no matter if the original problem is, is:

$$\begin{array}{ll} \underset{\lambda,\nu}{\text{maximize}} & g(\lambda, \nu) \\ \text{subject to} & \lambda_i \geq 0, \ i = 1, \ldots, m. \end{array}$$

The optimum solution of the Lagrange dual problem d^* is $d^* \leq p^*$.

3.3.3 Karush-Kuhn-Tucker conditions

Given a convex optimization problem with differentiable objective and constraint functions, the solutions that satisfy the Karush-Kuhn-Tucker (KKT) conditions are optimal for the primal and the dual problem. The optimal value of the Lagrange dual function is the optimal value of the primal optimization problem, $d^* = p^*$. Furthermore, if Slater's condition holds, then the KKT conditions are necessary and sufficient conditions for optimality.

For $\widetilde{x}, \widetilde{\lambda}, \widetilde{\nu}$ the KKT conditions are:

$$\begin{aligned} g_i(\widetilde{x}) &\leq 0, \quad i = 1, \ldots, m \\ h_i(\widetilde{x}) &= 0, \quad i = 1, \ldots, p \\ \widetilde{\lambda} &\geq 0, \quad i = 1, \ldots, m \\ \widetilde{\lambda} g_i(\widetilde{x}) &= 0, \quad i = 1, \ldots, m \\ \nabla f(\widetilde{x}) + \sum_{i=1}^{m} \widetilde{\lambda} \nabla g_i(\widetilde{x}) + \sum_{i=1}^{p} \widetilde{\nu} \nabla h_i(\widetilde{x}) &= 0. \end{aligned}$$

3.3.4 Support vector machines

A hyperplane is a set of the form

$$\{\mathbf{d}|\mathbf{w}^T\mathbf{d} = b\},$$

where $\mathbf{w} \in \mathbb{R}^n, \mathbf{w} \neq 0$, and $b \in \mathbb{R}$, see [36]. A hyperplane is the boundary of two halfspaces. Linear classifiers can be represented by a hyperplane, whereby an example $\mathbf{d}$ is classified with the sign function:

$$h(\mathbf{d}) = sgn(\mathbf{w}^T\mathbf{d} + b),$$

with the classes $c = \{+1, -1\}$.

Separable case Informally, the inductive bias of an SVM can be formulated as the objective of finding the hyperplane that maximizes the distance to the closest examples subject to the constraint that all training examples are correctly classified. For a linear separable training sample $\mathbf{S} = \{(\mathbf{d}_i, y_i),\ \mathbf{d}_i \in \mathbb{R}^n,\ y_i \in \{+1, -1\}\}$ with $i = 1, \dots, m$. The inequality constraints

$$y_i(\mathbf{w}^T\mathbf{d}_i + b) \geq 1$$

ensure that the set of feasible solutions only comprises consistent hypotheses. The resulting optimization problem is:

$$\begin{aligned} \underset{\mathbf{w},b}{\text{minimize}} \quad & f(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} \quad & y_i(\mathbf{w}^T\mathbf{d}_i + b) - 1 \geq 0,\ i = 1, \dots, m. \end{aligned}$$

Geometrically, $\mathbf{w}$ is orthogonal to the hyperplane and $|b|/\|\mathbf{w}\|$ is its perpendicular distance to the origin. The closest positive and negative examples have the perpendicular distances

$$\begin{aligned} dist_+ &= |1 - b|/\|\mathbf{w}\| \quad \text{and} \\ dist_- &= |-1 - b|/\|\mathbf{w}\|. \end{aligned}$$

The hyperplane $\mathbf{w}$ and its offset b are scaled so that the closest positive example $\mathbf{d}_+$ and the closest negative example $\mathbf{d}_-$ fulfill

$$\begin{aligned} \mathbf{w}^T\mathbf{d}_+ + b &= 1 \quad \text{and} \\ \mathbf{w}^T\mathbf{d}_- + b &= -1. \end{aligned}$$

Therefore, the margin $dist_+ + dist_-$ between these examples is $2/\|\mathbf{w}\|$, which is maximized by minimizing $\|\mathbf{w}\|^2$. Minimizing $\frac{1}{2}\|\mathbf{w}\|^2$ leads to the same solutions and is mathematically more convenient.

Introducing the Lagrange multipliers λ_i for the inequality constraints of the form $g_i(x) \geq 0$, the optimization problem can be stated as:

$$\underset{\lambda}{\text{maximize}} \inf_{\mathbf{w},b} \ \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \lambda_i \left(y_i(\mathbf{w}^T\mathbf{d}_i + b) - 1 \right),$$

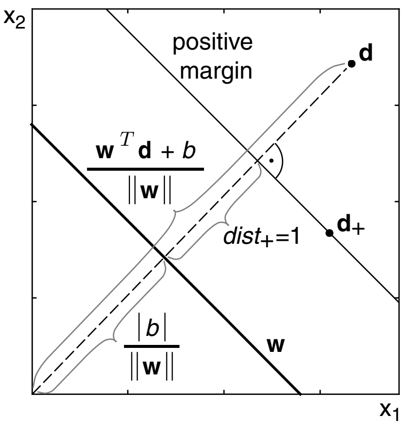


Figure 3.3: *Geometric representation of the hyperplane* **w** *and its positive margin in a two dimensional feature space.*

where the optimal solution of the dual is the solution of the primal problem. With the KKT conditions the same problem becomes:

$$\begin{array}{lll}\underset{\lambda}{\text{maximize}} & \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \lambda_i \left(y_i(\mathbf{w}^T\mathbf{d}_i + b) - 1 \right) & \\ \text{subject to} & y_i(\mathbf{d}_i^T\mathbf{w} + b) - 1 & \geq 0 \\ & \lambda_i & \geq 0 \\ & w_d - \sum_i \lambda_i y_i x_{i,j} & = 0 \\ & -\sum_i \lambda_i y_i & = 0, \end{array}$$

with $i = 1, \dots, m$ and $j = 1, \dots, n$. Due to the KKT conditions, the solution $\mathbf{w}$ is a linear combination

$$\mathbf{w} = \sum_{i=1}^{m} \lambda_i y_i \mathbf{d}_i,$$

where only the closest training examples, namely the examples with a perpendicular distance of one, influence the solution. These examples are so-called support vectors. The value of b can directly be computed by selecting any i with $\lambda_i > 0$ and set it into the KKT conditions [47].

Substituting the constraints to the objective function the optimization problem becomes:

$$\begin{array}{ll}\underset{\lambda}{\text{maximize}} & \sum_i \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{d}_i^T \mathbf{d}_j \\ \text{subject to} & \lambda_i \geq 0, \end{array}$$

with $i, j = 1, \dots, m$.

Non-separable case For the case, where the training examples are not separable by a linear hyperplane, no feasible solution can be found. Therefore the concept of a soft margin was proposed by Cortes and Vapnik [59] where misclassification is allowed but penalized. The penalty is introduced in the inequality constraints by slack variables ξ_i,

$$y_i(\mathbf{w}^T\mathbf{d}_i + b) \geq 1 - \xi_i,\ i = 1, \ldots, n.$$

Without going into the details, an important result of Cortes and Vapnik is that a C parameter can be introduced in the dual form of the resulting optimization problem, which virtually removes the slack variables:

$$\begin{aligned} \underset{\lambda}{\text{maximize}} \quad & f(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{d}_i^T \mathbf{d}_j \\ \text{subject to} \quad & 0 \leq \lambda_i \leq C \\ & \sum_{i=1}^{n} \lambda_i y_i = 0. \end{aligned}$$

Non-linear SVMs Based on the idea of projecting examples into a so-called inner-product space, SVMs are able to learn concepts that are not linear. In the inner-product space, the trained hyperplane is linear. A kernel function $K(\mathbf{d}_i, \mathbf{d}_j) = \phi(\mathbf{d}_i) \cdot \phi(\mathbf{d}_j)$ avoids computing the projection ϕ and the training examples are processed as dot products during the optimization procedure. The same applies, if a new example $\mathbf{d}$ is classified:

$$\begin{aligned} h(\mathbf{d}) &= sgn(\mathbf{w}^T\mathbf{d} + b) \\ &= sgn\left(\sum_i \lambda_i y_i \mathbf{d_i}^T \mathbf{d} + b\right), \end{aligned}$$

with $x_i \in$ "support vectors".

3.4 Evaluation of text classification solutions

A text classification solution m is a tuple comprising a feature engineering function α and a classifier h. The estimates of the effectiveness of a binary classifier are based on the confusion matrix, cf. Table 3.2. This table organizes the number of true positives (TP), false negatives (FN), false positives (FP), and true negatives (TN) reported in a testing procedure (e.g., leave-n-out) on a test sample. A sample has P positive and N negative examples, whereby PP examples are positively and NP examples negatively predicted.

The quality of m's predictions is quantified by measures summarized in Table 3.3, such as recall or precision. Some of these measures are known by other names: recall, for example, is also known as true positive rate or sensitivity; true negative rate is also known as specificity; false positive rate is also known as fallout; precision is also known as positive predictive value. We associate all

Table 3.2: *Confusion matrix. This matrix organizes the predicted along with the actual class values of the test sample that is used during an evaluation.*

		Actual class		
		Positive	Negative	Total
Predicted class	Positive	TP	FP	PP
	Negative	FN	TN	NP
	Total	P	N	

measures with the positive class by declaring the positive class as the class of interest (target class).

Without loss of generality, we use the term "effectiveness" as a generic term for such measures and presume the range $[0;1]$. A higher effectiveness $e(m, \mathbf{S})$ corresponds to a larger value of a measure if m is applied to a labeled test sample $\mathbf{S}$, and hence to a better prediction quality of m.

As mentioned above, when calculating one of the measures in Table 3.3, the target class is considered to be the positive class. In a two-class problem, the perspective from the negative class might also be important. In this context, the true negatives take the role of the true positives and the false negative the role of the false positives. For example, the recall Rec_{neg} of the negative class can be defined as

$$Rec_{neg} = \frac{TN}{TN + FP}.$$

Because of the false positives, the precision of the positive class explains the recall of the negative class and vice versa subject to the class balance.

Furthermore, for the interpretation of the results, one should always have the effectiveness of a classifier in mind that makes random predictions in terms of a uniformly distributed outcome over Y. The recall of this classifier is $1/|Y|$, and its precision is equal to the class balance. In contrast, in a multiclass setting with more than two classes, the precision of the class of interest is not sufficient to infer the recall of a specific counter class.

Table 3.3: *Common measures (name, event, statistic) that can be computed from the confusion matrix for estimating the effectiveness of a text classification solution m. The values $+1$ and -1 are associated with the positive and negative class labels. A prediction is denoted by $h(\mathbf{d})$ and the true value is denoted by $c(d)$.*

Name	**Event**	**Statistic**
Accuracy *Acc*	$\mathrm{P}(h(\mathbf{d}) = c(d))$	$\frac{TP+TN}{P+N}$
Precision *Prec*	$\mathrm{P}(c(d) = +1 \vert h(\mathbf{d}) = +1)$	TP/PP
Recall *Rec*	$\mathrm{P}(h(\mathbf{d}) = +1 \vert c(d) = +1)$	TP/P
True neg. rate *TNR*	$\mathrm{P}(h(\mathbf{d}) = -1 \vert c(d) = -1)$	$TN/\ N$
False pos. rate *FPR*	$\mathrm{P}(h(\mathbf{d}) = +1 \vert c(d) = -1)$	FP/N
False neg. rate *FNR*	$\mathrm{P}(h(\mathbf{d}) = -1 \vert c(d) = +1)$	FN/P
Rate of pos. predictions *RPP*	$\mathrm{P}(h(\mathbf{d}) = +1)$	$\frac{PP}{P+N}$
Rate of neg. predictions *RNP*	$\mathrm{P}(h(\mathbf{d}) = -1)$	$\frac{NP}{P+N}$
F-measure *F*		$2\frac{Prec \cdot Rec}{Prec+Rec}$

Chapter 4

Feature engineering for non-standard text classification tasks

In this chapter, we study methods for representing texts for several non-standard text classification tasks. A task is non-standard if classifiers combined with standard bag-of-words models suffer from symptoms either of low effectiveness or of low generalization capability. The quantization of these symptoms is a major hurdle in text classification. We therefore stress appropriate experimental setups and evaluation measures for each task.

Modeling text classification tasks without a representation that goes beyond the subject of texts becomes difficult for non-standard tasks with a topic-orthogonal class scheme. In this context, we point out the important role of writing style, which is discriminating for many non-standard text classification tasks. We introduce a novel form of representation, known as co-stems, which mainly captures the writing style in a text. Section 4.1 examines co-stems for a variety of tasks. In addition, this chapter explores features for the following tasks in detail:

- Web genre analysis (Section 4.2)
- information quality analysis (Section 4.3)
- language identification (Section 4.4)
- authorship verification analysis (Section 4.5)
- intrinsic plagiarism analysis (Section 4.6)

The experiments are conducted under laboratory conditions using standard text classification corpora to ensure the comparability and reproducibility of our results. A controlled environment is common and essential for feature engineering. Necessary conditions with respect to the features can be validated and descriptive statistics calculated. This is important for qualifying the discriminative power and the interaction of features with respect to the given task. Descriptive statistics are typically used for scaling and normalizing features. Evaluations under laboratory conditions are also useful for parameter tuning in order to approach the goals in information filtering and retrieval, where one might prefer a high precision for spam filtering, and one might prefer a high

Table 4.1: *Overview of information filtering tasks. The tasks which are analyzed in this section are tagged with the ✓-symbol.*

Task	Description	Reference
Age group analysis	Determine the age of the author who wrote *d*.	[262]
Authorship attribution	Determine the author of *d*, given a set of authors.	[281]✓
Authorship verification	Determine if *d* is written by more than one author.	[153]
Gender analysis	Determine the gender of the author who wrote *d*.	[262]✓
Genre analysis	Determine the genre of *d*, given a set of genres.	[295]✓
Information quality analysis	Determine whether *d* is of high quality.	[177]✓
Language identification	Determine the language of *d*.	[99]
Sarcasm analysis	Determine whether *d* is sarcastic.	[308]
Sentiment analysis	Determine the sentiment expressed in *d*.	[224]✓
Spam detection (email, webpage)	Determine whether *d* is spam or non-spam.	[26, 218]✓
Topic categorization	Determine the topic of *d*.	[161]✓
Vandalism analysis	Determine whether *d* is vandalized.	[237, 235]

recall for patent retrieval. A drawback is that it is difficult to observe if a text representation misleads a classifier, which affects the generalization capabilities. The employed corpora are not representative of the real situation in the Web; class schemes and balances are unknown, and labeling noise occurs.

4.1 Co-stems in non-standard text classification tasks

Identifying relevant, interesting, high quality, or humorous texts in wikis, emails, and blogs is the tedious job of every information seeker. Algorithmic information filtering [104] simplifies this process by finding those texts in a stream or a collection that fulfill a given information interest. Current information filtering technology mostly relies on text classification where the classes describe the information interests. Usually the text representations are content-based, although various filtering tasks are characterized by their intricate combination of content and style.

In this section, we evaluate whether the untapped potential of a style representation is substantial. We propose a model that encodes both (1) text content and (2) text style in the form of word stems and word co-stems respectively. To draw a clear and comprehensive picture of the underlying effects and their importance we resort to a straightforward vector representation. We consider the computational simplicity of this representation as a useful contribution, and to the best of our knowledge the co-stem representation has not been proposed or investigated in this respect. Also the number and heterogeneity of information filtering tasks that are compared in this section goes beyond existing evaluations. In particular, we analyze the tasks in Table 4.1 that are marked with ticks (✓) and refer to the relevant literature. In this table, *d* denotes a text that is extracted

Table 4.2: *Different co-stems for the words "timelessly" and "timeless" resulting from different stemming algorithms.*

Stemming Algorithm	Co-stem	Stem	Co-stem	Co-stem	Stem	Co-stem	Reference
Porter, Lancaster, Krovetz	–	timeless	ly	–	timeless	–	[233, 221, 158]
Lovins	–	time	lessly	–	tim	eless	[189]
3-Truncation	–	tim	elessly	–	tim	eless	
rev. 3-Truncation	–	timeles	sly	–	timel	ess	

from an email, a wiki page, a blog entry, or a webpage, depending on the task in question.

4.1.1 Co-stems

Co-stems are constructed by the following operation: given a word its stem is computed first, and then the residuals of the word without its stem are taken as co-stems. For example, consider the words "timeless" and "timelessly" along with the application of different stemming algorithms, shown in Table 4.2.

Stems are the output of a stemming algorithm, which is "[. . .] a computational procedure that reduces all words with the same root (or, if prefixes are left untouched, the same stem) to a common form, usually by stripping each word of its derivational and inflectional suffixes" [189]. A root is the base form of a word and cannot be reduced without losing its identity. An inflectional suffix changes the grammatical role of a word in a sentence, it indicates gender, number, tense, etc. A derivational suffix is used for word-formation. For example, the word "timelessly" has the inflectional suffix "ly" and the derivational suffix "less".

A word can have at most three co-stems, namely the part before, after, and inside the stem. Depending on the employed stemming algorithm, a co-stem can be a single affix or a combination of affixes. Note that most stemming algorithms are language-dependent, and that some stemming algorithms regard a stem as one or more root morphemes plus a derivational suffix (the Lovins stemmer does not).

Derivational affixes and inflectional suffixes depend on the part of speech. Typical derivational affixes are for example for nouns "-ion, -ment, -ance" and for verbs "en-, be-, de-, -ify, -en, -ate, -ize". Therefore, analyzing co-stems provides not only information about the usage of derivations and inflections, but also implicit information about the usage of the part of speech.

4.1.2 Analyzing co-stems

The general setting in our evaluation is as follows: Given a text classification task, the Lovins stemming algorithm [189] computes the stems of an extracted text. The algorithm uses a list of 297 suffixes and strips the longest suffix from a word; hence the resulting co-stems in this study are suffixes.

Since the goal is to capture the writing style of a text, we enhance the set of co-stems with stop words and punctuation. A text d is represented by a vector $\mathbf{d}$, where each dimension specifies the frequency of its associated token. A token can be a word, a stem, or a co-stem. We apply as classification technologies a generative approach, as well as a discriminative approach, namely naïve Bayes (NB) and linear support vector machines (SVM). Naïve Bayes is a generative classifier that learns a model of the joint probability, $\mathrm{P}(\mathbf{D}, Y)$. Its decision rule selects the most likely class y by calculating $\mathrm{P}(y \mid \mathbf{d})$ using the Bayes rule. The SVM is a discriminative classifier that learns a direct mapping from $\mathbf{D}$ to Y by following the structural risk-minimization principle.

Each corpus in this study is specific for its respective field and accepted as comparable standard resource. The corpora consist of several categories that are regarded as classes or labels in a classification setting. Here, each corpus is broken down into two classes by randomly selecting two categories and ignoring the additional ones. The number of examples in each category is kept assessable.

"The Blog Authorship Corpus" from Schler et al. [262] is used, which contains blog entries from blogger.com organized by female and male bloggers and by topical categories.

We compiled a corpus with the texts extracted from Wikipedia articles. It distinguishes between "featured" (high quality) and "non-featured" articles that were randomly chosen from the entire Wikipedia without restricting to selected domains.

Moreover, the well-known "20 Newsgroups" corpus with Usenet articles is employed. Therefore, we have sampled articles from the top-level category "comp.*" (computer-related discussions, comprising 5 categories) and from the top-level category "rec.*" (recreation and entertainment, comprising 4 categories).

Another corpus is the "7-Web genre collection" from Santini [257] which consists of webpages categorized in blogs, eshops, FAQs, online newspaper front pages, listings, personal home pages, and search pages.

Furthermore, we took course pages and non-course pages, used in the co-training experiments by Blum and Mitchell [28], from "The 4 Universities Dataset".

Finally, the "SpamAssassin public email corpus" and the "Cornell Movie Review Dataset" by Pang and Lee [222] are employed.

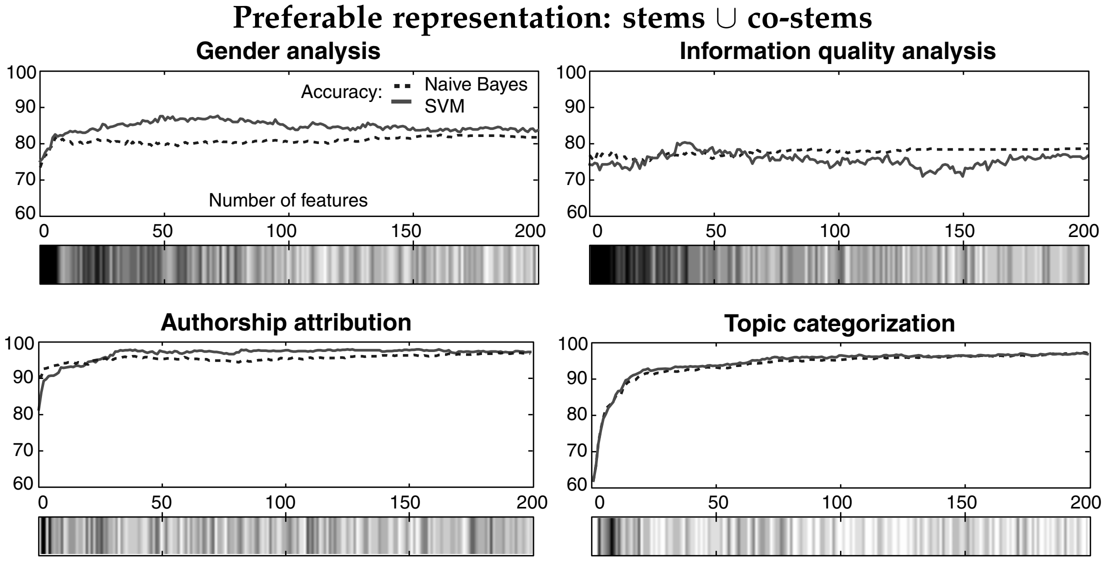


Figure 4.1: *Task-specific discrimination analysis of co-stems. Each plot shows the classification accuracy over the number of employed features for a given task. The curves correspond to the naïve Bayes classifier (dotted) and SVM (solid) respectively. The striped bars illustrate whether a stem (white) or a co-stem (black) is chosen by information gain as the i-th feature. The results are obtained from a 10-fold cross-validation and the averaged value is reflected by the shade of gray. A dark left area indicates the superiority of co-stems.*

Effectiveness comparison Table 4.3 compares the classification effectivenesses of words, stems, co-stems, and co-stems combined with stems. The symbols ○ and ● indicate a statistically significant improvement and degradation respectively, compared with the bag-of-words model in a paired T-test with 0.05 significance. For each precision, recall, F-measure, and area under ROC curve (P, R, F, AUC) the average is given, weighted by the class distribution. The best solution of a task in terms of the F-measure is shown bold. The effectivenesses are averaged over ten repetitions of a 10-fold cross-validation. The table also shows further details of the used corpora. Since we consider only binary classification tasks, we randomly select two categories for those corpora that cover more than two categories.

Co-stems are effective in gender analysis, while the combination of co-stems and stems leads to the best classification result. The combination leads also to the best results in information quality analysis and authorship attribution, and it is able to compete in topic categorization. For genre analysis (e-shop vs private home page) and genre analysis (course vs non course) the effectivenesses of the representation based on stems is comparable with the best solution. Finally, the standard bag-of-words model performs best in spam detection and sentiment analysis.

Influence of co-stems To understand the influence of stems and co-stems in information filtering, Figure 4.1 illustrates the feature importance characteristics

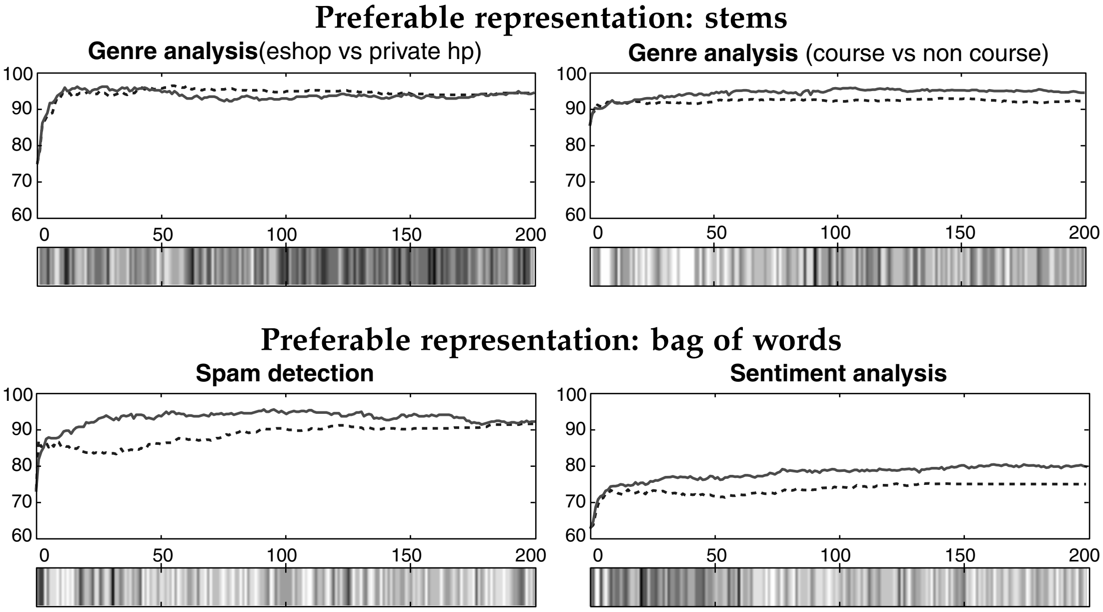


Figure 4.2: *Continuation of Figure 4.1.*

for each task. The illustrations show the 10-fold cross-validation accuracies of the SVM and NB classifiers if the top k features (stems and co-stems) are used. The top k features are computed by the information gain criteria on the training split in each fold. The striped bars below the figures illustrate the preference between stems (white) and co-stems (black) according to the information gain criterion among the top 200 features.

The frequent occurrence of co-stems in all tasks among the top 200 features emphasize the discriminative power of co-stems. Each task has its own characteristics that are shown by the classification accuracy and the gray scale. A dark left area indicates a superior impact of co-stems with respect to the classification effectiveness in the specific task, which can be observed in particular for gender analysis and information quality analysis. Co-stems are valuable within tasks where the texts to be filtered typically originate from a specific writer or group of uniform writers. Examples are authorship attribution and information quality analysis, where a high quality Wikipedia article is edited by a group of writers who are likely to share style elements.

4.1.3 Summary

Each non-standard text classification task has its own characteristics in terms of the importance of co-stems. For the tasks gender analysis, information quality analysis, and authorship attribution the combination of stems and co-stems leads to a statistically significant improvement compared with bag of words. We provide evidence for the discriminative power of co-stems by setting up experiments with accepted corpora and by analyzing and illustrating the distribution of the top discriminating features.

Table 4.3: *Classification effectiveness. ○, ● indicates a statistically significant improvement or degradation with respect to the std. bag-of-words representation in a paired T-Test with 0.05 significance. Each measure is class-dependent and the weighted average result is given. The best solution of a task in terms of the F-measure is shown in bold. Continued in Table 4.4.*

	Naïve Bayes				SVM			
Representation	**P**	**R**	**F**	**AUC**	**P**	**R**	**F**	**AUC**
Task: Gender analysis *Corpus*: 400/400 blog entries written by different male/female bloggers. *Source*: "The Blog Authorship Corpus" [262] with 681,288 blog entries from 19,320 bloggers on blogger.com.								
Bag of words	0.72	0.71	0.71	0.78	0.70	0.70	0.70	0.74
Stems ∪ Co-stems	0.82 ○	0.82 ○	0.82 ○	0.90 ○	0.87 ○	0.86 ○	**0.86** ○	0.91 ○
Stems	0.77 ○	0.77 ○	0.77 ○	0.84 ○	0.80 ○	0.80 ○	0.80 ○	0.85 ○
Co-stems	0.83 ○	0.83 ○	**0.83** ○	0.90 ○	0.86 ○	0.85 ○	0.85 ○	0.91 ○
Task: Information quality analysis. *Corpus*: 255/255 "featured" (high quality) and "non-featured" articles. *Source*: The english version of Wikipedia.								
Bag of words	0.79	0.78	0.78	0.87	0.84	0.83	0.83	0.87
Stems ∪ Co-stems	0.80 ○	0.80 ○	**0.80** ○	0.88 ○	0.87 ○	0.87 ○	**0.87** ○	0.91 ○
Stems	0.80 ○	0.80 ○	**0.80** ○	0.86	0.81 ●	0.81 ●	0.81 ●	0.84 ●
Co-stems	0.78	0.78	0.78	0.87	0.86 ○	0.85 ○	0.85 ○	0.91 ○
Task: Authorship attribution *Corpus*: 357/481 blog entries from one author/from all other authors. *Source*: The engineering category from"The Blog Authorship Corpus" [262].								
Bag of words	0.98	0.97	**0.97**	1.00	0.98	0.98	0.98	1.00
Stems ∪ Co-stems	0.97	0.97	**0.97**	1.00	0.99 ○	0.99 ○	**0.99** ○	1.00
Stems	0.96 ●	0.96 ●	0.96 ●	1.00	0.98 ○	0.98 ○	0.98 ○	1.00
Co-stems	0.96 ●	0.95 ●	0.95 ●	1.00	0.95 ●	0.94 ●	0.94 ●	0.99 ●
Task: Topic categorization. *Corpus*: 1,000/800 messages from the (top-level) newsgroups computer-related discussions/recreation and entertainment. *Source*: The well-known "20 Newsgroups" with 20,000 Usenet articles.								
Bag of words	0.98	0.98	**0.98**	1.00	0.97	0.97	0.97	0.99
Stems ∪ Co-stems	0.98	0.98	**0.98**	1.00	0.98 ○	0.98 ○	**0.98** ○	0.99
Stems	0.98	0.98	**0.98**	1.00	0.98 ○	0.98 ○	**0.98** ○	1.00
Co-stems	0.83 ●	0.82 ●	0.82 ●	0.90 ●	0.88 ●	0.88 ●	0.88 ●	0.93 ●

Table 4.4: *Continuation of Table 4.3*

	Naïve Bayes				SVM			
Representation	**P**	**R**	**F**	**AUC**	**P**	**R**	**F**	**AUC**
Task: Genre analysis (e-shop vs private home page). *Corpus*: 200/200 webpages from eshops/personal home pages. *Source*: The "7-Web genre collection" [257] with 1,400 webpages.								
Bag of words	0.96	0.96	0.96	0.99	0.94	0.94	**0.94**	0.98
Stems ∪ Co-stems	0.95 •	0.94 •	0.94 •	0.98 •	0.94 ○	0.93	0.93	0.98
Stems	0.97 ○	0.97 ○	**0.97** ○	0.99	0.94	0.93	0.93	0.98
Co-stems	0.87 •	0.85 •	0.85 •	0.95 •	0.88 •	0.86 •	0.86 •	0.94 •
Task: Genre analysis (course vs non course) *Corpus*: 230/821 webpages about courses/non-courses. *Source*: The subset of "The 4 Universities Dataset" used in the co-training Experiments [28].								
Bag of words	0.94	0.94	**0.94**	0.98	0.93	0.91	0.91	0.98
Stems ∪ Co-stems	0.92 •	0.92 •	0.92 •	0.96 •	0.91 •	0.88 •	0.89 •	0.98
Stems	0.93 •	0.93 •	0.93 •	0.97 •	0.93 ○	**0.92** ○	0.92 ○	0.98
Co-stems	0.88 •	0.89 •	0.88 •	0.91 •	0.91 •	0.90	0.90 •	0.95 •
Task: Spam detection *Corpus*: 160/320 spam/non-spam emails. *Source*: The "SpamAssassin public email corpus" with 1,397 spam and 2,500 non-spam emails. `http://spamassassin.apache.org`								
Bag of words	0.92	0.92	**0.92**	0.97	0.95	0.94	**0.94**	0.98
Stems ∪ Co-stems	0.92	0.91 •	0.91 •	0.96 •	0.93 •	0.91 •	0.91	0.98 •
Stems	0.92	0.91 •	0.91 •	0.96 •	0.94 •	0.92 •	0.93	0.98 •
Co-stems	0.89 •	0.89 •	0.89 •	0.95 •	0.93 •	0.93 •	0.93 •	0.96 •
Task: Sentiment analysis *Corpus*: 1,000/1,000 positve/negative movie reviews. *Source*: The "Cornell Movie Review Dataset" [222] with 1,000 positve and 1,000 negative reviews.								
Bag of words	0.80	0.80	**0.80**	0.88	0.85	0.85	**0.85**	0.91
Stems ∪ Co-stems	0.76 •	0.76 •	0.75 •	0.84 •	0.84 •	0.83 •	0.83 •	0.91
Stems	0.81	0.80	**0.80**	0.89	0.82 •	0.82 •	0.82 •	0.89 •
Co-stems	0.63 •	0.62 •	0.62 •	0.68 •	0.72 •	0.72 •	0.71 •	0.79 •

4.2 Core-vocabularies in Web genre analyses

This chapter contributes to Web genre analysis and core-vocabularies. A core-vocabulary is a group of words that indicates a functional aspect of a text. We propose concentration characteristics of genre-specific core-vocabularies as generalizable and efficiently computable features for genre analysis. In this connection we introduce methods for mining tailored core-vocabularies, as well as particular statistics as a means for a sensible feature quantization. Special focus is put on the generalization capability of Web genre classification, for which we present evaluation measures and, for the first time, a quantitative analysis.

4.2.1 Web genre analysis

The genre of a Web document provides information related to the document's form, purpose, and intended audience. Documents of the same genre can address different topics and vice versa, and several researchers consider genre and topic as orthogonal concepts. Though this claim does not hold without exceptions, genre information attracted much interest as positive or negative filter criterion for Web search results.

Early work in automatic genre classification dates back to 1994, where Karlgren and Cutting [139] presented a feasibility study for a genre analysis based on the Brown corpus. Later on followed several publications investigating different corpora, using more intricate or less complex retrieval models, stipulating other concepts of genre, or reporting on new applications [143, 346, 282, 11, 241, 69, 85].

Genres *on the Web* have been investigated since 1998, for example, by Bretan et al. [41]. Table 4.5 compiles research that received attention: the table lists the basis of the analysis, the genre palette Q, and the text representation $\mathbf{d}$. The underlying use case is a genre-enabled Web search. The approaches from Crowston and Williams [62], Roussinov et al. [250] and Dimitrova et al. [71] are not included since the authors provided suggestions rather than a technical specification about their genre retrieval models.

Though the undoubted potential of an automatic genre identification for web-pages, models for genre could not convince in the Web retrieval practice by now. The reasons for this are threefold. First, also observed by Santini [258], the proposed genre classifier technology is corpus-centered: their application within Web retrieval scenarios shows a significant degradation of the classification effectiveness, rendering the technology largely useless for genre-enabled Web search. Second, the existing genre retrieval models are computationally too expensive to be applied in an ad-hoc manner. Third, there is no genre palette that fits for all users and all purposes. Ideally, the users should be able to adapt

Table 4.5: *Research in the field of automatic genre classification for Web-based corpora and digital libraries. An important use case is the development of a richer retrieval result representation in the search interface. From Stein, Meyer zu Eißen, and Lipka [295].*

Author **Analysis basis**	**Web genre palette** Q	**Text representation d**
Bretan et al. (1998) user study with 102 interviewees	private, public/commercial, journalistic, report, other texts, interactive, discussion, link collection, FAQ, other listing	simple part-of-speech features, emphatic and down-toning expressions, relative number of digits, average word length, number of images, proportion of links
Lee and Myaeng (2002) 7615 documents	FAQ, home page, reportage, editorial, research article, review, product specification	genre-specific core-vocabulary
Rehm (2002) 200 documents	hierarchy with three granularity levels for academic home pages	HTML metadata, presentation-related tags, linguistic features
Meyer zu Eissen and Stein (2004) user study with 286 interviewees, 800 documents	article, discussion, shop, help, personal home page, non-personal home page, link collection, download	word-frequency class, part-of-speech, genre-specific core-vocabulary, other close-classed word sets, text statistics, HTML tags
Kennedy and Shepherd (2005) 321 documents	personal, corporate, organizational	HTML tags, phone, email, presentational tags, CSS, URL, link, script, genre-specific core-vocabulary
Boese and Howe (2005) 342 documents	abstract, call for papers, FAQ, sitemap, job description, resume, statistics, syllabus, technical paper	readability scales, part-of-speech, text statistics, HTML tags, bow, HTML title tag, URL, number types, closed world sets, punctuation
Lim et al. (2005) 1224 documents	home page, public, commercial, bulletin, link collection, image collection, simple list, input, journalistic, research, official material, FAQ, discussion, product specification, informal	part-of-speech, URL, HTML tags, token information, most frequent function words, most frequent punctuation marks, syntactic chunks
Freund et al. (2006) 800 documents	best practice, cookbook, demo, design pattern, discussion, documentation, engagement, FAQ, manual, presentation, problem, product page, technical, technote, tutorial, whitepaper	bag of words
Santini (2007) 1400 documents	blog, listing, eshop, home page, FAQ, search page, online newspaper front page	most frequent English words, HTML tags, part-of-speech, punctuation symbols, genre-specific core-vocabulary
Santini (2007) 2480 documents	[as before]	text type analysis plus a combination of layout and functionality tags

a genre classifier to their information need, for examples, by labeling documents as being of an interesting genre or not.

From the mentioned deficits the first one is the most severe: put in a nutshell, the existing Web genre classification models generalize insufficiently. Also the second deficit is crucial since it makes the important use case of a genre-enabled Web search unattractive for users who expect a result list from a search engine by the press of a button. We demonstrate how so-called core-vocabularies are employed to construct effective text classification solutions for this task.

Insufficient generalization capabilities The authors of the approaches listed in Table 4.5 report on classification results for the correct assignment of genre classes. The obtained cross-validated accuracies are surprisingly high, reaching from 75% with $|Q| = 16$ genre labels in [173] up to 90% with $|Q| = 7$ genre labels in [167]. These and similar results are achieved with rather small training corpora, containing between several hundred and a few thousand documents.

Let m_1 be the genre classification solution trained on corpus D_1, and let m_2 be the solution trained on corpus D_2. With respect to the common genre labels of two concrete classification models, Santini [258] investigated the generalization capability of m_1 to corpus D_2 and vice versa.[1] It turns out that the precision decreases by more than an order of magnitude.

The classification knowledge that is operationalized within m_1 can only be exported to a corpus D_2 if the model captures the *intensional semantics* of the concept "genre". The intensional semantics of a genre classification model can be understood as its capability to comply with the extensional semantics of genre in different worlds, say, as its capability to correctly classify documents from different corpora. If so, the model provides a high generalization capability resulting from a moderate inductive bias, cf. Section 2. Most of the Web genre models listed in Table 4.5 have a weak bias.

High computational efforts Table 4.5 lists a wide range of feature types commonly used in Web genre analysis:

Presentation-related features Frequency counts for figures, tables, paragraphs, headlines, captions. HTML-specific analysis regarding colors, hyperlinks, URLs, or mail addresses.

Simple text statistics Frequency counts for clauses, paragraphs, delimiters, question marks, exclamation marks, and numerals.

Special closed-class word sets and core-vocabularies Use of currency symbols, help phrases, shop phrases, calendar, or countries.

[1] Santini [258] uses the term "exportability" in this connection. Actually, she measured the agreement between m_1 and m_2, which is a particular facet of the generalization capability [309].

Word-frequency class analysis Use of special, common, or misspelled words.

Part-of-speech analysis Frequency counts for nouns, verbs, adjectives, adverbs, prepositions, or articles.

Syntactic group analysis Use of tenses, relative clauses, main clauses, adverbial phrases, or simplex noun phrases.

Following Stein, Meyer zu Eißen, and Lipka [295], the effort of computing the mentioned features is between linear time in the text length, for simple frequency counts, and ranges up to cubic effort and higher for the parsing of syntactic groups. The usefulness and, even more important, the cost-benefit ratio of these features with respect to a reliable genre analysis is unclear. Hence, the researchers who build genre retrieval models tend to include a feature instead of leaving it out.

The feature selection is shifted to the learning algorithm, which identifies and weights the most discriminating features with respect to the training sample. This strategy is acceptable if the training sample is plentiful and sensibly distributed with respect to the classification task; both requirements are not fulfilled here. The construction of training corpora is expensive, as the small sample sizes in the first column of Table 4.5 show. Moreover, the different user and task-specific genre palettes and the impracticality of estimating the document distribution on the Web are the reasons that very little can be stated about the a-priori probabilities of genres. The combination of rich feature models with small training corpora is crucial in two respects: it compromises generalization capability and makes the learning process sensible to the training sample. A way out is the use of few features with a coarse domain.

4.2.2 Core-vocabularies

For the set D of documents let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be an exclusive genre partition of D; i.e., $\bigcup_{C \in \mathcal{C}} C = D$ and $\forall C_i, C_{j,j \neq i} \in \mathcal{C} : C_i \cap C_j = \emptyset$. For a genre $C \in \mathcal{C}$, let T_C denote the core-vocabulary specific for C. Similar to Broder [43], we argue that T_C is composed of navigational, transactional, structural, and informative words. The combination, distribution, presence or absence of these words encode a considerable part of the genre information.

Navigational words appear in labels of hyperlinks and in anchor tags of webpages. Examples are: “Windows”, “Mac”, or “zip” in download sites, links to “references” in articles.

Transactional words appear in sites that interact with databases, and manifest in hyperlink labels, forms, and button captions. Examples are: “add to shopping cart”, “proceed to checkout” in online shops, buttons labeled “download” on download pages.

Structural words appear in sites that maintain meta-information such as time and space. Examples include meta-information of posts in a discussion forum ("thread", "replies", "views", parts of dates) and words that appear in addresses on home pages ("address", "street").

Informative words appear not in functional HTML elements but imply functionality though. Examples include "kB" or "version" on download sites, "price" or "new" on shopping sites; and "management", "technology", or "company" on commercial sites.

Vocabulary construction The words in T_C are both predictive for C and frequent in C. Words with such characteristics can be identified in $\mathcal{C}$ with approaches from topic categorization research, in particular Popescul's method and the weighted centroid covering method [232, 164, 165, 290]. For mining a genre-specific core-vocabulary both methods must be adapted: they do not quantify whether a word is representative for C; a deficit that can be repaired without compromising the efficient $O(m \log(m))$ runtime of the methods, where m designates the number of words in the dictionary [292].

Concentration measures In the simplest case, the relation between T_C and a document d can be quantified by computing the fraction of d's words from T_C, or by determining the coverage of T_C by d's words. However, if genre-specific vocabulary tends to appear concentrated in certain places on a webpage, this characteristic is not reflected by the mentioned features, and hence, it cannot be learned by a classifier. Examples for webpages in which genre-specific core-vocabulary appears concentrated: private home pages (e.g., address vocabulary), discussion forums (e.g., words from mail headers), and non-personal home pages (e.g., words related to copyright and legal information). The following two statistics quantify two different vocabulary concentration aspects:

Maximum Word Concentration Let $d \in D$ be represented as a sequence of words, $d = \{w_1, \ldots, w_m\}$, and let $W_i \subset d$ be a text window of length l in d starting with word i, say, $W_i = \{w_i, \ldots, w_{i+l-1}\}$. A natural way of measuring the concentration of words from T_C in different places of d is computing the following function for each W_i:

$$\kappa_{T_C}(W_i) = \frac{|W_i \cap T_C|}{l}, \qquad \kappa_{T_C}(W_i) \in [0,1]$$

The overall concentration is defined as the maximum word concentration:

$$\kappa^*_{T_C} = \max_{W_i \subset d} \kappa_{T_C}(W_i), \qquad \kappa^*_{T_C} \in [0,1]$$

Gini Coefficient In contrast to the κ_{T_C} statistic, which quantifies the word concentration strength within a text window, the Gini coefficient can be utilized for quantifying to which extent a genre-specific core-vocabulary is distributed unequally over a document. Again, let W_i be a text window of size l sliding over d. The number of genre-specific words from T_C in W_i is $\nu_i = |T_C \cap W_i|$. Let A denote the area between the uniform distribution line and the Lorenz curve of the distribution of ν_i, and let B denote the area between the uniform distribution line and the x-axis. The Gini coefficient is defined as the ratio $g = A/B, g \in [0,1]$. A value of $g = 0$ indicates an equal distribution; the closer g is to 1 the more unequal ν_i is distributed.

Concentration measures capture the distribution of different subsets of a document's words with respect to their position in the document. These subsets, denoted as core-vocabularies here, as well as their concentration analysis, form the basis for non-linear features that cannot be constructed by the state-of-the-art learning technology.

4.2.3 Analyzing Web genre models

This section addresses the evaluation-related issues of Web genre analysis. We discuss approaches for improving the generalization capability and propose statistics for quantifying this property for genre classification models. Each experiment is repeated and averaged using ten randomly drawn samples of the respective number of training examples; the applied learning algorithm is a support vector machine and the text representations vary. Our empirical analysis illustrates the theoretical observation from above: the stronger the structural bias of a classification model is, the higher is its generalization capability.

Corpora The analysis is based on the Web genre corpora "KI-04" with eight Web genre classes by Meyer zu Eißen and Stein [201], denoted as A, and the "7-Web genre collection" by Santini [257], denoted as B.[2] These corpora are sketched in Table 4.5, row 4 and row 8.

Feature engineering We improve a classifier's generalization capability by restricting its structural bias. In practice, this goal is achieved by (1) reducing the number of features, (2) reducing the number of values a feature can take, and (3) replacing weak features by discriminative features. The proposed concentration measures, maximum concentration and Gini coefficient of core-vocabulary distributions, impose one feature per genre class, resulting in eight features for a document of a collection with eight genre classes. In comparison with a standard genre model, the number of features introduced by these concentration measures is orders of magnitude smaller. The following text representations are examined:

[2] KI-04 can be downloaded from `http://www.webis.de/research/corpora`. In the experiments the extended version of this corpus (1 200 webpages) is used.

GenreVSM The vector space model using $tf \cdot idf$ term weighting scheme, comprising about 3 500 features.

GenreVoc A core-vocabulary model based on the core-vocabulary analysis as introduced in Subsection 4.2.2, comprising a total of 26 features.

GenreBasic A basic genre model based only of HTML features, link features, and character features, comprising a total of 54 features.

GenreRich A rich genre model based on the features of GenreBasic along with part-of-speech and vocabulary concentration features, comprising a total of 98 features.

GenreRichNoVoc The GenreRich model without the vocabulary concentration features, comprising a total of 72 features.

Measuring generalization capability In what follows, the concepts classifier agreement and export accuracy will be defined; the notation is adapted from Turney [309]. Informally, these concepts quantify the classification effectiveness, the impact of classifier variation, and the impact of corpus variation.

Classifier agreement Let D be a document set organized according to a genre scheme Q. Moreover, let $\alpha_1 : D \rightarrow \mathbf{D}_1$ and $\alpha_2 : D \rightarrow \mathbf{D}_2$ be two text representations and let $m_1 = (\alpha_1, h_1)$ and $m_2 = (\alpha_2, h_2)$ be two genre classification solutions. Then the agreement of the classifiers h_1 and h_2 is defined as follows:

$$agree(h_1, h_2) \; := \; P\left(h_1(\mathbf{d_1}) = h_2(\mathbf{d_2})\right),$$

where $\mathbf{d}_1$ and $\mathbf{d}_2$ are representations of the same document $d \in D$ generated by α_1 and α_2 respectively.

That is, the classifier agreement is the probability that two genre classification solutions make the same decision on the genre of a document. $\alpha_1 = \alpha_2$ can hold: the two solutions rely on the same text representation but differ with respect to their machine-learning settings. In particular, h_1 and h_2 can result from training on different samples while using the same classifier type. In this important analysis case, the classifier agreement quantifies the training sample sensibility of a genre classification solution.

Export accuracy Let $D_1 \subset D$ and $D_2 \subset D$ be two document sets organized according to the genre palettes Q_1 and Q_2, $Q_1 \cap Q_2 \neq \emptyset$. Moreover, let α be a function that computes the text representations $\mathbf{D}_1 \subset \mathbf{D}$ and $\mathbf{D}_2 \subset \mathbf{D}$, and let (α, h) be a classification solution for D_1. Then the export accuracy of the classification solution (α, h) with respect to D_2 is defined as follows:

$$e_{h,D_2} \; := \; P\left(h(\mathbf{d}_2) = c^*\right),$$

where $\mathbf{d}_2 \in \mathbf{D}_2$ is the representation of a document $d_2 \in (D_2 \setminus D_1)$ with genre class $c^* \in (Q_1 \cap Q_2)$.

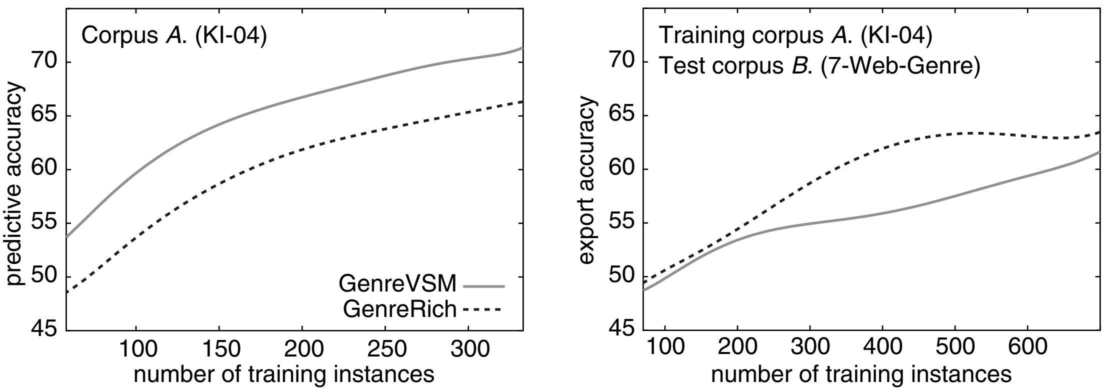


Figure 4.3: *Accuracy (left) and export accuracy (right) of the text representations GenreVSM and GenreRich, depending on the size of the training set, which is always drawn from corpus A (KI-04). The accuracy is estimated on a test sample of corpus A, while the export accuracy is estimated on a test sample of corpus B (7-Web genre collection).*

That is, the export accuracy is the probability that the assigned genre of a document of an external corpus is correct. Note, the export accuracy is affected by the homogeneity of the training corpus. The export accuracy of a genre classification solution (α, h) with respect to D_2 quantifies whether the combination of D_1, α, and h_1 captures the gist of the genre classes in $Q_1 \cap Q_2$. Only if the document set D_1 is representative, if α is sensible, and if h generalizes sufficiently, the classification solution will perform acceptably for the documents in D_2.

Experiment 1: Export accuracy The presumably most important property of a Web genre classification solution is a high export accuracy. In this connection, the left plot in Figure 4.3 shows the accuracy of the representations GenreVSM and GenreRich, trained on and applied to documents of corpus A containing 1 200 documents. The right plot shows the export accuracy of these representations with respect to corpus B containing 600 documents, with $Q_A \cap Q_B = \{$shop, personal home page, link list$\}$. In both plots the x-axis shows the sample size of the training set taken from corpus A; the y-axis shows the corresponding accuracy on corpus A (left plot) and the export accuracy on corpus B (right plot).

The GenreVSM model achieves a significantly higher accuracy than the GenreRich model (see Figure 4.3, left plot); with respect to the sample size both show the same consistency characteristic. We explain the high accuracy of GenreVSM with its higher training sample sensibility, which is beneficial in homogeneous corpora. Even under a successful cross-validation test the accuracy and the export accuracy will considerably diverge.

A corpus may be homogeneous because of the following reasons:

- The corpus is compiled by a small group of editors who share a very similar understanding of genre.

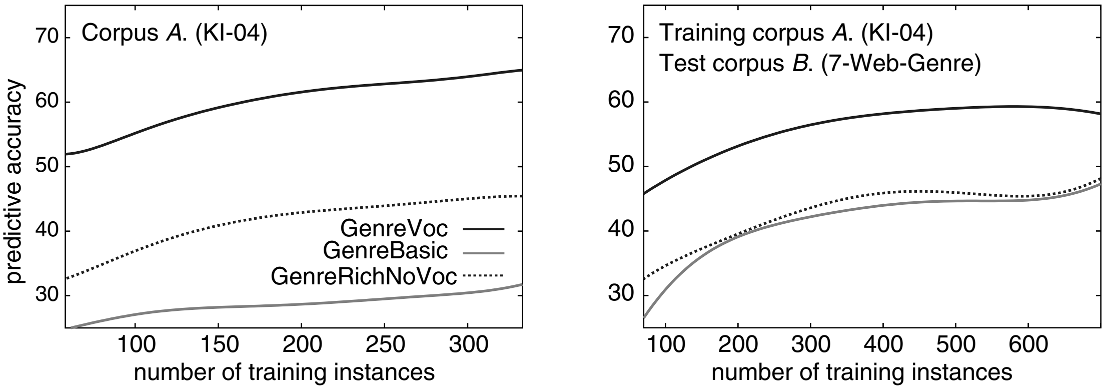


Figure 4.4: *Accuracy (left) and export accuracy (right) of the text representations GenreVoc, GenreRichNoVoc, and GenreBasic, using the same settings as in the experiments shown in Figure 4.3.*

- The editors introduce subconsciously an implicit correlation between topic and genre.
- The editors collect their favored documents only.
- The editors rely on a single search engine whose ranking algorithm is biased towards a certain document type.

Corpus homogeneity is unveiled when analyzing the export accuracy, which drops significantly (by 21%) for the GenreVSM model (see Figure 4.3, right plot). For the GenreRich model the export accuracy drops only by 8%. The robustness of the GenreRich model is a consequence of its small number of features, which is more than an order of magnitude smaller compared with the GenreVSM model.

The plots shown in Figure 4.4 quantify also the drop in export accuracy (left plot → right plot), but analyze different classification model variants whose feature sets are subset of the GenreRich model:

- The GenreVoc model shows a small drop in the export accuracy, which is rooted in the fact that the core-vocabulary has a small, acceptable corpus dependency.
- For the GenreRichNoVoc model, the export accuracy remains pretty constant. The reasons for this stability are the small hypothesis space and a small corpus dependency of the features.
- For the GenreBasic model, the export accuracy is significantly higher than the accuracy. We explain this behavior with the high discriminative power of the HTML features and link features with respect to the genre classes shop, personal home page, and link list.

Experiment 2: Classification agreement Figure 4.5 shows results from an agreement analysis for classifiers of the GenreVSM model and the GenreRich model. The x-axis denotes the size of the training set, which is always drawn

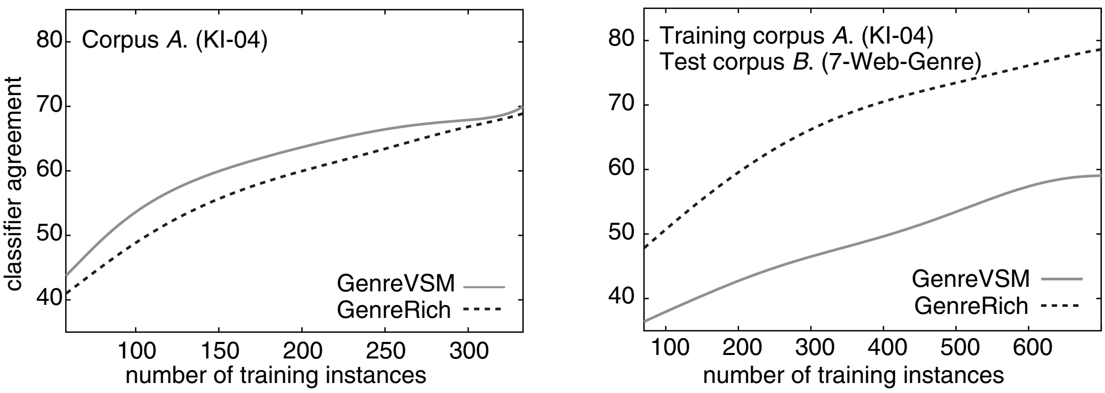


Figure 4.5: *Classifier agreement of the text representations GenreVSM and GenreRich, depending on the size of the training set, which is always drawn from corpus A. In the left plot the agreement is analyzed on corpus A, and in the right plot the agreement is analyzed on corpus B.*

from corpus *A* (KI-04). As expected, both plots show the monotonous characteristic of the classifiers subject to the training set size.

In the left plot of Figure 4.5, it can be observed that the agreement of both classifiers is quite similar, although the model formation bias of the GenreVSM model is weaker than the bias of the GenreRich model. Again, this behavior can be explained by the homogeneity of the corpus. Nevertheless, the situation is different if the classifier agreement is analyzed on a test corpus different from the training corpus (see the right plot in Figure 4.5): the agreement of classifiers under the GenreRich model is much better than the agreement of classifiers under the GenreVSM model. I.e., classifiers under the GenreVSM model are corpus-specific (overfitted) whereas classifiers under the GenreRich model are not, they provide a much higher generalization capability.

4.2.4 Summary

Most of the existing genre text representations exploit high-level features, such as part of speech, tailored text statistics, or information about the document structure. Apart from the high computational effort, a negative consequence is that the resulting genre classification solution tends to generalize unsatisfactorily. Especially because of the last point, classification solutions for genre analysis did not convince in the Web retrieval practice. Our research addresses this issue as it provides formal means for measuring the generalization capability. We also propose a feature type for text representations that quantifies the concentration of genre-specific core-vocabularies in a document, and that has the potential of improving the generalization capability of existing genre classification models. Our analysis shows that this new feature type is successful in this respect.

4.3 *N*-grams in information quality analyses

The automatic assessment of information quality is becoming a key factor in information retrieval. If this is possible in its generality is an open question since the quality of a text is subjectively perceived: it depends on the users' contexts, their expectations, and their prior knowledge. Wikipedia provides a controlled environment, where high-quality articles are labeled as "featured", after being run through an extensive human peer review process. The Wikipedia community characterizes featured articles among others as well-written, comprehensive, well-researched, neutral, and stable. In Wikipedia, but of course in every retrieval situation, the declaration of an article's quality helps users to focus on the valuable information sources. This section focuses on the automatic identification of featured articles in Wikipedia.

Several researchers develop metrics that are suitable for capturing quality indicators but demanding in computational respects: Zeng et al. [351] and McGuinness et al. [198] computed an article's trustworthiness using revision history features and citation features. Stvilia et al. [298] developed metrics that are based on edits, editors, links, article length, age, and readability indices. Brandes et al. [37] indicated structural parameters of the edit network. Stein and Hess [296] and Adler and de Alfaro [1] developed authorship-based quality ratings, which concern the amount of the authors contributions in an article and a reputation estimate. Hu et al. [122] took the reviewership into account, which relies on the assumption that unedited content is reviewed by an author who edits the respective article.

Blumenstock [29] proposed the word count of an article; it is a simple metric that works significantly better than several of the aforementioned metrics. His approach, the classification of articles with more than 2 000 words as featured, is performing well for an unbalanced corpus with a large amount of short articles. Our experiments show an effectiveness decline for balanced corpora, as well as when only articles with 1 500–2 500 words are used.

Assessing the quality of a text can be operationalized as a text classification task. In the case of classifying texts from all possible sources (e.g., wikis, blogs, mailing lists, and comment boards), structural features such as the word-count mentioned are not generalizing anymore. Our approach opens the powerful toolbox developed for tackling authorship problems. We employ various trigram text representations along with a classifier to identify featured articles. In particular, we examine their robustness and generalizability in domain transfer experiments. Especially character trigram vectors, which are not yet considered in information quality research, are a promising representation: they are comparable to word counts in simplicity but gain a higher discriminability.

Table 4.6: *The most discriminative character trigrams.*

Rank	Trigram
1-20	ing|ng |, a|at |e, |er | an|ed |d a| be ter|s a| re|as |ted|g a|tha|n t| a |ly
21-40	to | th|nd |. a|on |sed|t t|eve|tin|er, as|r, |d s|th |red| on|ear| to|n a|he
41-60	at|or |d t|s, |g t|, w|for|s w|s f| fo e a|s t|r t|est| ha|din|hat| wi| di|all
61-80	s s|d o|e s|s r|by |ver|d, |ve | in|ore rin|ere|s c|the|in |and|st |d b|t a|wit
81-100	s. |en |e o| ma|ion|e w|s b| by|ved|ut no|ain|d w| wh|’s |her| de|e t|e e|was

4.3.1 Automatic information quality assessment

Starting point is the classification task “Is an article featured or not?”. For this purpose we apply, again, linear support vector machines (SVM) and naïve Bayes (NB). Our study deals with writing-style-related representations of articles and their binarizations: character trigram vectors and part-of-speech (POS) trigram vectors.

An n-gram vector of a text d is an $\ell1$-normed numeric vector, where each dimension specifies the frequency of its associated n-gram in d. An n-gram in turn is a substring of n tokens of d, where a token can be a character, a word, or a POS tag. The vector is binarized if the occurrence or non-occurrence of an n-gram is counted as 1 and 0.

POS n-gram vectors and character n-gram vectors are writing-style-related since they capture intrinsics of an author’s text synthesis traits. POS n-grams unveil sentence construction preferences; character n-grams unveil preferences for sentence transitions, as well as the utilization of stop words, adverbs, and punctuation—all of which are important authorship indicators [281]. To illustrate how writing style matters with respect to our classification task, Table 4.6 compiles the most discriminative character trigrams, ranked by information gain on our evaluation corpora. It should be noted that authorship indicators are more important than topic indicators such as word stems.

4.3.2 Evaluation

The English Wikipedia domains Biology and History are used as sources for the compilation of two corpora: given the extracted plain texts with more than 800 words per article from a domain, all available featured texts and the same number of non-featured texts are added to the respective corpus. Altogether 180+180 articles belong to Biology, and 200+200 articles belong to History. We run three kinds of experiments:

Table 4.7: *Identification performance for featured articles, within and across domains (P/R/F $\sim$ Precision/Recall/F-measure). Maximum F-measure values are shown in bold.*

Representation	**Classifier**	**Identification of featured articles (P/R/F)**	
Cross-Validation.		*within Biology*	*within History*
bin char trigram	SVM	0.966 / 0.961 / **0.964**	0.888 / 0.955 / **0.920**
bin POS trigram	SVM	0.949 / 0.933 / 0.941	0.889 / 0.925 / 0.907
word count	SVM	0.755 / 0.600 / 0.669	0.874 / 0.870 / 0.872
bag of words	NB	0.832 / 0.989 / 0.904	0.860 / 0.950 / 0.903
Domain Transfer.		*History → Biology*	*Biology → History*
bin char trigram	SVM	0.800 / 0.978 / **0.880**	0.886 / 0.855 / **0.870**
bin POS trigram	SVM	0.799 / 0.883 / 0.839	0.898 / 0.790 / 0.840
word count	SVM	0.772 / 0.733 / 0.752	0.878 / 0.830 / 0.853
bin bag of words	SVM	0.800 / 0.889 / 0.842	0.930 / 0.665 / 0.776

Cross-validation Evaluate a classifier h by 10-fold cross-validation within a single domain. Rationale of the experiment is to minimize the influence of topical discrimination, which can occur if articles of more than one domain are shuffled.

Domain transfer Construct a classifier h with articles from a source domain (training), and apply h to a different target domain (test). The experiments, denoted as "*source domain → target domain*", show both the potential of transferring relations about information quality across domains and the generalization ability of h.

Length sensitivity Apply a classifier h constructed within the domain transfer experiment to the three sets that contain articles with less than 1 500 words, with 1 500–2 500 words, and with more than 2 500 words. The interesting questions are:

- Is the article length sufficient for robust feature computation?
- Is it sensible to combine a word-count-based classifier with an n-gram-based classifier?

Table 4.7 (Cross-Validation and Domain Transfer) and Table 4.8 (Length Sensitivity) summarize the results of the trigram vector representations and, as baselines, the bag-of-words representation and the word count approach. Only the best performing representations, binarized or non-binarized, and classifiers, SVM or NB, are mentioned in the tables. Here, binarized trigram vectors outperform the non-binarized: about +0.2 averaged F-measure in the cross-validation experiments and +0.3 in the domain transfer experiments. The binarized character trigram vectors are most effective. The length sensitivity analysis shows that the

Table 4.8: *Identification effectiveness for featured articles across domains, broken down with respect to article lengths (F ∼ F-measure). Classification technology are SVMs. ⊥ indicates Precision=Recall=0.*

Representation	**Identification of featured articles (F)**		
	< 1500 words	1500–2500 words	> 2500 words
Length Sensitivity.	*History → Biology*		
	1% featured articles	22% featured articles	77% featured articles
bin char trigram	1.000	0.860	0.885
word count	⊥	0.677	0.852
Length Sensitivity.	*Biology → History*		
	3% featured articles	8% featured articles	89% featured articles
bin char trigram	⊥	0.316	0.888
word count	⊥	⊥	0.905

combination of a word-count-based classifier with an *n*-gram-based classifier achieves no improvement.

The word count discrimination rule Blumenstock [29] suggested a length discrimination rule: articles with more (less) than 2 000 words are classified as featured (non-featured), yielding an accuracy of 0.96 for an unbalanced corpus (ratio 1:6, featured : non-featured).

Figure 4.6 shows the probability densities over word count for our balanced corpora; also here, the 2 000 word threshold is close to the optimum discrimination rule. Yet, we achieve only an accuracy of 0.79 within Biology and 0.89 within History via the length discrimination rule. In contrast, the binarized character trigram vector representation combined with an SVM yields an accuracy of 0.96 within Biology and 0.92 within History.

4.3.3 Summary

We examine the character trigram feature, originally applied for writing-style analysis [281], which has not been considered for the information quality assessment in Wikipedia so far. We study existing research and new solutions that combine different text representations and learning algorithms. Altogether, the combination of a linear SVM with a binarized character trigram text representation has convincing properties: it yields for featured articles a high identification effectiveness, even across domains, it handles unstructured text, and it is computationally efficient.

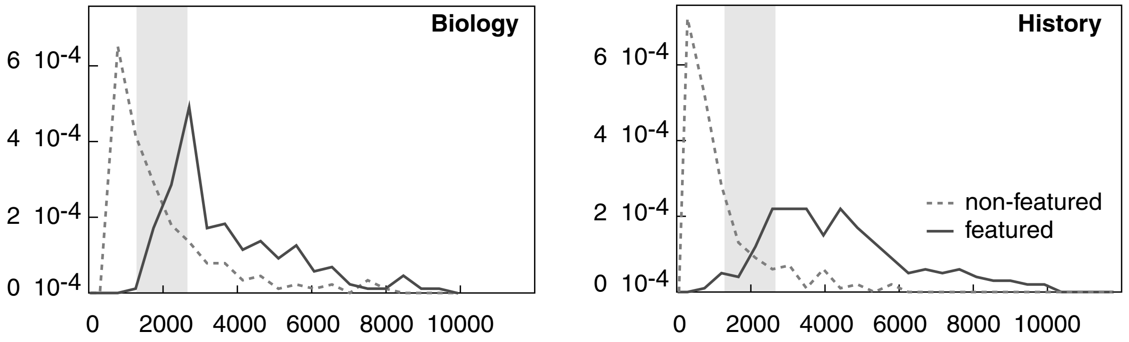


Figure 4.6: *Probability density over absolute word count.*

4.4 N-grams in short text language identification

The focus in research on language identification has been usually on analyzing full documents, i.e., on reasonably long and well formulated texts. This is considered a solved problem. For cross-language applications, identifying the language on short texts becomes important, which is rarely studied.[3] This section shows the potential and reliability of common language-identification approaches applied to very short, query-style texts. Because of the lack of large annotated multi-language query corpora, we based our experiments on news headlines of the Reuters CV1 and CV2 collection and single words, which we extracted from bilingual dictionaries.

Automatic language identification on written texts, also known as language detection or language recognition, is a text classification task. The most distinguished related work is based on statistical learning algorithms and lexical text representations, particularly n-grams [77, 51, 320, 299, 274]. Dictionary-based approaches, concerning words as lexical representation, are discussed by Dunning [77] and by Rehurek and Kolkus [245]. Non-lexical representations, such as phoneme transcriptions or the rate of compression, have been studied by Berkling et al. [22] and Teahan [305].

4.4.1 Use cases

The difficulty of a Cross Language Information Retrieval (CLIR) system is to find relevant documents across language boundaries. This induces the need for a CLIR system to be capable of doing translations between documents and queries. If the system has to handle more than one language for queries or documents, it additionally needs to be able to detect the language of a text. This is necessary for correctly translating the query or the document into the language of the respectively other. Queries, instead, are rarely formulated as full sentences and are usually very short (typically 2-4 words for web search). Recent systems that have participated at CLEF detect the language of queries by

[3] Two month after the publication of our study [99] a similar results have been reported on different corpora by Vatanen et al. [319].

applying tools intended for long texts, e.g., [219]. Therefore, we ask how well these approaches work on short texts such as single words?

4.4.2 *N*-gram-based language identification

As short, query-style texts provide too little data for approaches that are based on multiple words or full sentences, we focus on methods based on character *n*-grams, for short *n*-grams. An *n*-gram consists of *n* sequential characters; usually its relative occurrence in a text is determined for building a text representation. A language identification method based on this were proposed, for example, by Cavnar and Trenkle [51]. The following classification methods are studied in the evaluation section:

Frequency-rank Following the observation that each languages has characteristic *n*-grams that appear frequently, this model compares the *n*-gram frequency-rank of an unseen text with the frequency-ranks of reference texts for different languages. The text is then attributed to the language with the most similar rank according to an out-of-place measure. As this measure is problematic because of the few entries in the frequency-rank list of short texts, we normalize the ranks in our implementation to values between 0, for the most frequent, and 1, for the least frequent *n*-gram.

Naïve Bayes The naïve Bayes classifier uses the conditional probabilities of observing *n*-grams in a text to deduce the class probability that a text is written in a given language.

Markov The idea is to detect the language of a text via the probability of observing its character sequences in a specific language. The classifier selects the language that maximizes this probability [320, 77].

Vector space A text is represented as a vector of its *n*-gram frequencies. A language is represented by the mean vector of all example texts in this language. To classify the language that belongs to a text, the most similar language in the vector space is determined by the cosine similarity.

4.4.3 Evaluation

Corpora All the proposed methods need to be trained on reference texts. We use the English texts in the Reuters collections CV1 [171] and the Danish, German, Spanish, French, Italian, Dutch, Norwegian, Portuguese and Swedish texts from CV2. Table 4.9 shows the distribution of the individual languages among the 1 102 410 texts.

Table 4.9: *Distribution of languages in the Reuters corpus and in dictionary terms. From Gottron and Lipka [99].*

Corpus	da	de	en	es	fr	it	nl	no	pt	sv
Reuters	11 184	116 209	806 788	18 655	85 393	28 406	1 794	9 409	8 841	15 731
Dictionaries	–	3 463	12 391	3 260	1 153	2 432	–	–	501	–

Experiment To study the influence of the length of n-grams we varied n between one and five characters. The relatively short and noisy Reuters headlines are retained for classification. They are on average 45.1 characters and 7.2 words long, thus, longer than an average query on the Web. The titles frequently contain named entities (“*Berlusconi* TV faces legal cliffhanger”) or numerical values (“Dollar General Q2 *$0.24* vs *$0.20*”). These entities and a lack of stop words render the headlines a quite suitable set of short, query-alike texts for language identification. For the evaluation of single words, we obtain words from small, bilingual dictionaries; from English to French, German, Spanish, Italian and Portuguese. We extract the words, which are unambiguous from a language point of view, i.e., which exist in only one language. This results in a total of 20 048 words of on average 8.1 characters, see Table 4.9 for details about individual languages.

The algorithms are implemented from scratch and trained on the Reuters articles. For the frequency-rank approach, we additionally use a readily trained implementation of the original algorithm, which we include in the evaluation process as LC4J[4]. We use each of the algorithms to detect the languages of the previously unused Reuters headlines and the words obtained from the dictionaries.

Table 4.10 shows the accuracies for detecting the language of the Reuters headlines and the dictionary entries across all algorithms and all settings for n. But, the values of LC4J need to be treated carefully: in many cases the algorithm could not detect any language at all. This might be because the language models provided with the implementation are too sparse for short texts. The values given here are solely based on those cases where the language identification resulted in an output. When taking into account the unclassified texts, the accuracy drops drastically to 39.24% for the headlines and to 30.33% for the dictionary words.

The poor effectiveness of the Markov process and our own frequency-rank implementation for higher values of n can be explained with data sparseness, too. The accuracy of Markov drops due to the high number of n-grams that are not seen during training and an unequal language distribution in the training sample. The frequency-rank approach instead suffers from the sparseness of n-grams in the query-like texts, which results in skewed rankings. Even with the normalized ranking, the effectiveness drops for larger values of n.

[4] `http://olivo.net/software/lc4j/`

Table 4.10: *Accuracy of language classifiers (in %). From Gottron and Lipka [99].*

Data	Method	1-grams	2-grams	3-grams	4-grams	5-grams
Headlines	Naïve Bayes	87.90	95.01	98.52	99.40	**99.44**
	Multinomial	65.42	90.08	97.63	99.17	99.22
	Markov	10.28	85.87	73.13	4.50	0.00
	Frequency-rank	6.07	14.90	59.93	25.91	3.47
	Vector space	54.68	47.21	61.04	69.67	75.37
	LC4J (where successful)	–	–	67.72	–	–
Dictionaries	Naïve Bayes	52.26	64.40	73.49	79.13	**81.61**
	Multinomial	35.65	57.04	68.27	75.74	77.88
	Markov	19.95	57.34	55.14	21.52	2.95
	Frequency-rank	12.32	24.04	42.82	23.25	6.70
	Vector space	29.99	33.98	44.28	52.73	59.23
	LC4J (where successful)	–	–	49.93	–	–

The best performing method for short texts is the naïve Bayes classifier and its multinomial variation without the class distribution normalization. For larger values of n both variations perform remarkably good and achieve an accuracy close to 100% on the headlines. This observation holds also when studying the accuracy for individual languages. On a language level, the accuracy varies between 99.71% for Italian and 96.52% for Norwegian. The misclassifications of Norwegian headlines are mostly assigned to Danish. In general, the Scandinavian languages tend to be confused more than other languages. A similar observation is made for dictionary words of Latin-based languages. Here the most mistakes occur between Spanish, Portuguese and Italian.

4.4.4 Summary

Applying n-grams for identifying the language of short, query-style texts is demonstrated, for the first time, to be effective. Comparing different approaches based on n-grams, it turns out, that naïve Bayes classifiers perform best on very short texts and even on single words.

4.5 Function words in authorship verification analyses

In an authorship verification problem one is given writing examples from an author A, and one is asked to determine whether or not each text has been written by A. Koppel and Schler [153] compared the usage of function words in A's writing examples and in the unseen text with in a so-called unmasking approach. In this section, we report to what extend unmasking is applicable if the *a priori given* writing examples are noisy, that is, if not all of them stem from A.

4.5.1 Authorship verification

The heart of an authorship verification analysis is the quantization of an author's writing style along with the identification of anomalies via outlier classification. For outlier classification, one is given a target class for which a certain number of examples exist; objects outside the target class are called outliers. A one-class classifier tells apart outliers from target class members. Actually, the set of outliers can be much bigger than the target class, and an arbitrary number of outlier examples could be collected. Hence a one-class classification problem may look like a two-class discrimination problem; however, there is an important difference: members of the target class can be regarded as representatives for their class, whereas one will not be able to compile a set of outliers that is representative for some sort of "non-target class". This fact is rooted in the enormous number and diversity of outliers. Put another way, solving a one-class classification problem means to learn the concept of the target class in the absence of discriminating features.[5] Within authorship verification the target class comprises writing examples of a certain author A, whereas each piece of text written by another author B, $B \neq A$, is an outlier.

The major part of existing research focuses on models for writing-style quantification. Research related to authorship verification divides into the following areas: (1) models for the quantification of writing style, using classical measures for text complexity and grading level assessment [52, 121, 147, 101, 64, 86, 349], as well as author-specific stylometric analyses [284, 283, 154, 153, 152], (2) technology for outlier analysis and machine learning [303, 304, 240, 192], and (3) meta-knowledge processing. Regarding the last area we refer to techniques for knowledge representation, deduction, and symbolic knowledge processing [252, 285].

Koppel and Schler [153] introduced a new approach, namely unmasking, to determine with a high accuracy if a set of writing examples is a subset of the target class. The unmasking approach does not solve the one-class classification problem for a single text but requires two sets of texts, D_1 and D_2. D_i, $i = 1, 2$,

[5] In rare cases, knowledge about outliers can be used to construct representative non-target sample. Then a standard discrimination approach can be applied.

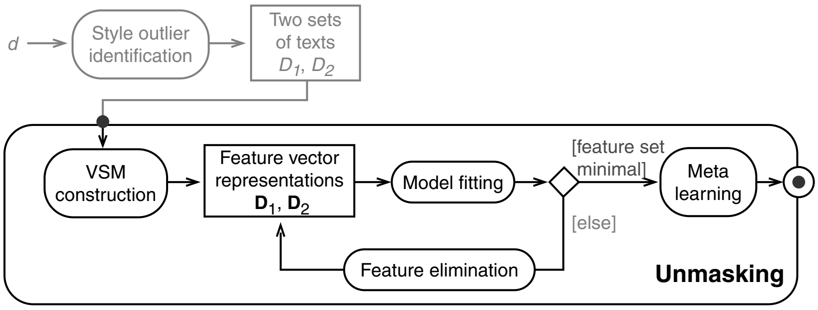

d
Style outlier identification
Two sets of texts D_1, D_2
VSM construction
Feature vector representations $\mathbf{D}_1$, $\mathbf{D}_2$
Model fitting
[feature set minimal]
Meta learning
[else]
Feature elimination
Unmasking

Figure 4.7: *Unmasking: given are two sets D_1 and D_2 of outlier texts and target texts. The basic idea is to measure the separability of $\mathbf{D}_1$ vs $\mathbf{D}_2$ when the style model is successively impaired. From Stein, Lipka, and Meyer zu Eißen [293].*

have to be a clean set of texts, that is, all texts must either stem from the target class or not. As this is not often the case in practice, we study the effectiveness of unmasking if this condition is violated.

4.5.2 Function words

Unmasking operationalizes a function-word analysis; it analyzes an author's usage of function and stop words. These words are characterized by a high frequency in texts and conditional independence with respect to topics. Early studies reported that function words are discriminative in authorship tasks, cf. Section 2.1.

The unmasking approach requires a closer look to understand its rationale. At first, the text sections in D_1 and D_2 are represented under a reduced vector space model, designated as $\mathbf{D}_1$ and $\mathbf{D}_2$. The 250 words with the highest frequency in $D_1 \cup D_2$ form the initial feature set. Unmasking happens in the following steps, cf. Figure 4.7:

(1) Model fitting Training of a classifier that separates $\mathbf{D}_1$ from $\mathbf{D}_2$. Koppel and Schler [153] implemented a 10-fold cross-validation experiment to determine the accuracy of a linear SVM.

(2) Impairing Elimination of the most discriminative features with regard to the model obtained in Step (1) and the new construction of the collections $\mathbf{D}_1$, $\mathbf{D}_2$ under the impaired representations of the texts. Koppel and Schler [153] reported on convincing results by eliminating the six most discriminating features; however, this heuristic depends on the text length.

(3) Go to Step 1 until the feature set is sufficiently reduced. Typically, about 5-10 iterations are necessary.

(4) Meta learning Analyze the degradation in the quality of the model fitting process: if after the last impairing step the sets $\mathbf{D}_1$ and $\mathbf{D}_2$ can still be separated with a small error, assume that d_1 and d_2 stem from different

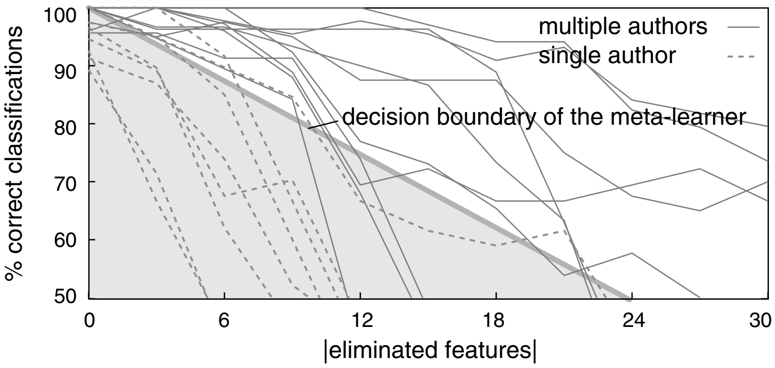


Figure 4.8: *Unmasking at work: each line corresponds to a comparison of two papers, where a solid (dashed) line belongs to papers from two different authors (the same author). From Stein, Lipka, and Meyer zu Eißen [293].*

authors. Figure 4.8 shows a characteristic plot where unmasking is applied to short papers of 4-8 pages.

In this step, a meta-learner is trained to distinguish the unmasking learning curves. The input vectors of the meta-learner comprise the following elements: the accuracy in iteration i, the Δ-*Acc* to iteration $i-1$, the Δ-*Acc* to iteration $i-2$, and a class label "multiple authors" or "single author". This meta-learner is also realized by a linear SVM.

The rationale of unmasking is the following: two sets of texts sections, D_1 and D_2, constructed from two different texts d_1 and d_2 of the same author can be told apart easily if a vector space model is chosen. The vector space model considers all words in $d_1 \cup d_2$, and hence it includes all kinds of open and closed class word sets. If only the 250 most frequent words are selected, a large fraction of them will be function words and stop words. Among these 250 most frequent words a small number does the major part of the discrimination job. These words capture differences that result from genre, purpose, topic and the like. By eliminating them, one approaches step by step the distinctive and subconscious manifestation of an author's writing style. After several iterations the remaining features are not powerful enough to discriminate two texts of the same author. By contrast, if d_1 and d_2 stem from different authors, the remaining features will still quantify significant differences between the impaired representations $\mathbf{D}_1$ and $\mathbf{D}_2$ of the two sets of sections D_1 and D_2.

4.5.3 Evaluating unmasking in authorship verification

Corpus Our test corpus contains scientific documents written in German. Basis of the corpus are dissertation and habilitation theses from the following fields: philosophy, psychology, sociology, medical science, historical science, and law. From the original theses all explicitly declared citations are removed, and clippings of about 10 000 words are extracted. These clippings represent the documents written by a single author; 10% of these documents are used to

Table 4.11: *The table illustrates the potential of unmasking in authorship verification, when D_1 and D_2 are constructed from d. It is important to note that if D_1 or D_2 is empty, unmasking cannot be employed; these cases are not listed. The baseline hypothesis is: if two non-empty sets could be constructed, d is written by different authors.*

Different Authors	**Construction of D_1 and D_2**	D_1	D_2	Potential of unmasking
perfect construction; unmasking can only decrease effectiveness				
true	*perfect*	clean	clean	can decrease recall ✓
imperfect construction (noise); unmasking can in-/decrease effectiveness				
false	*imperfect*	clean	noisy	can increase precision ✓
true	*imperfect*	noisy	noisy	can decrease recall ✓
true	*imperfect*	noisy	clean	can decrease recall ✓
true	*imperfect*	clean	noisy	can decrease recall ✓

construct impure documents by inserting 4 to 8 text sections of 500 words from foreign authors.

Experiment: Unmasking Under laboratory conditions, the two sets D_1 and D_2 within authorship verification are clean, i.e., D_1 does not contain elements from another author, while D_2 does not contain elements from the author of D_1's elements. This scenario rarely occurs in practice, cf. Table 4.11. In fact, the majority of possible scenarios where unmasking can be applied involves the risk that the recall of authorship verification decreases whereas the potential of increasing the precision only occurs in one scenario.

To evaluate unmasking under realistic conditions, "noisy" sets D_1 and D_2 are needed. For each document d in our test corpus, D_1 and D_2 are constructed as follows:

Decomposition The document d is chunked into text sections of equal length. Remark: Meyer zu Eißen and Stein [202] proposed an additional sentence detection for this step. Also, a more sensible interpretations of structural boundaries (chapters, paragraphs) is possible, which should consider special text elements such as tables, formulas, footnotes, or quotations [247]. The detection of topic boundaries has a significant impact on the usefulness of a decomposition [58]. Graham et al. [100] even tried identifying stylistic boundaries.

Set compilation, D_1 and D_2 A two-class classifier with a text representation based on style features proposed by Stein and Meyer zu Eißen [291] and Meyer zu Eißen and Stein [202] is trained for distinguishing between singular and multiple authorship. This classifier achieves a recall of 0.80 for both, the class of outlier sections and the class of target sections.[6]

[6] The classifier is artificially constructed. It cannot be trained for real-world authorship

Table 4.12: *Classification results for the unmasking strategy compared with the minimum-risk baseline.*

Impurity	**minimum risk**			**unmasking**		
θ	*Prec*	*Rec*	*F*	*Prec*	*Rec*	*F*
0.20	0.12	1.00	0.56	0.73	0.90	0.82
0.30	0.20	1.00	0.60	1.00	0.93	0.97
0.40	0.18	1.00	0.59	1.00	0.87	0.94

Table 4.12 shows an effectiveness comparison between the baseline (minimum risk) and unmasking:

Minimum risk outputs "multiple authors" if $|\mathbf{D}_1| \geq 1$ and $|\mathbf{D}_2| \geq 1$; "single author" otherwise, and

Unmasking outputs the result ("single author" or "multiple authors") of the described unmasking approach.

As expected, the recall in authorship verification decreases if unmasking is employed but the precision increases, cf. Table 4.11. When comparing the F-measures, unmasking seems to decrease the overall effectiveness compared with the baseline. However, this interpretation is critical. The baseline does not identify any single-author documents. The precision increase is of great importance in this context and unmasking clearly outperforms the minimum risk strategy.

4.5.4 Summary

The analysis of function words with the unmasking approach of Koppel and Schler [153] has a high risk to decrease the recall of finding documents that are written by multiple authors, and it has a low chance to increase the precision of authorship verification. Even though, we report that unmasking makes the best of this situation: it improves, on average, the precision by about 60% relative to a simple baseline, and the recall is lowered by about 50%, "only".

4.6 Style markers in intrinsic plagiarism analyses

Research in the field of automatic text plagiarism detection focuses on the development of algorithms that compare suspicious documents against a collection of reference documents. Recent approaches perform well in identifying copied

verification because of missing representative training examples. Usually, a tailored one-class classification approach should be applied here which learns from target examples only, cf. Section 4.6.

or modified foreign sections; however, they assume a closed world where a reference collection is given.

This section investigates the question whether plagiarized text sections can be detected by a computer program even if no reference collection is provided, for example, if the foreign sections stem from a book that is not available in digital form. We call this problem class intrinsic plagiarism analysis. The term plagiarism refers to *text* plagiarism, that is, the use of another author's information, language, or writing, when done without proper acknowledgment of the original source. Plagiarism *analysis* refers to the unveiling of text plagiarism.

The contributions of this section are threefold. (1) We organize the algorithmic building blocks of intrinsic plagiarism analysis and show how to transform intrinsic plagiarism problems via stylometry into authorship verification problems. (2) We employ "unmasking" to post-process weak or imperfect stylometry results. (3) We operationalize an analysis chain consisting of document chunking, style model computation, one-class classification, and meta learning, and we provide a plagiarism corpus with about 3 000 cases to evaluate the potential of our ideas. The meta learning combines heuristic voting with unmasking.

4.6.1 Intrinsic plagiarism analysis

Research on automated external plagiarism detection presumes a closed world where a reference collection D is given; the question is whether a given document d contains a section s that has a high similarity to a section s_i of a document $d_i \in D$. Since D can be extremely large, possibly the entire indexed part of the Web, the main research focus is on efficient search technology: near-similarity search and near-duplicate detection [42, 118, 23, 112, 114, 341], tailored indexes for near-duplicate detection [84, 23, 44], or similarity hashing techniques [150, 129, 97, 287, 288].

Intrinsic plagiarism analysis is closely related to authorship verification: the goal of intrinsic plagiarism analysis is to identify possibly plagiarized sections by analyzing a document with respect to "undeclared" changes of writing style. Similarly, in an authorship verification problem one is given writing examples from an author A, and one is asked to determine whether or not a text with doubtful authorship is also written by A. Intrinsic plagiarism analysis and authorship verification are one-class classification problems, whereas intrinsic plagiarism analysis can be understood as a more general form of the authorship verification problem where one is given a single document d only, and the question is whether or not d contains sections from other authors.

4.6.2 Operationalizing intrinsic plagiarism analysis

Plagiarism detection can be operationalized by decomposing a document into "natural" sections, such as sentences, chapters, or topically related blocks, and analyzing the variance of stylometric features for these parts following [294]. We organize this process into three stages:

Pre-analysis stage A knowledge-based "impurity" assessment gives us hints regarding the size and the distribution of suspicious sections in a document d, and where a tailored decomposition strategy is chosen. These decisions influence the construction of a model for writing-style quantification in the next stage.

Stylometric analysis stage A style model is constructed and style outliers are identified with respect to the typical writing style in d. Based on this analysis, an instance of an intrinsic plagiarism task is transformed into an instance of an authorship verification task.

Post-processing stage The result of the stylometry analysis stage is further analyzed with additional knowledge and meta-learning technology. Main objective in this stage is the improvement of the analysis' overall precision and recall.

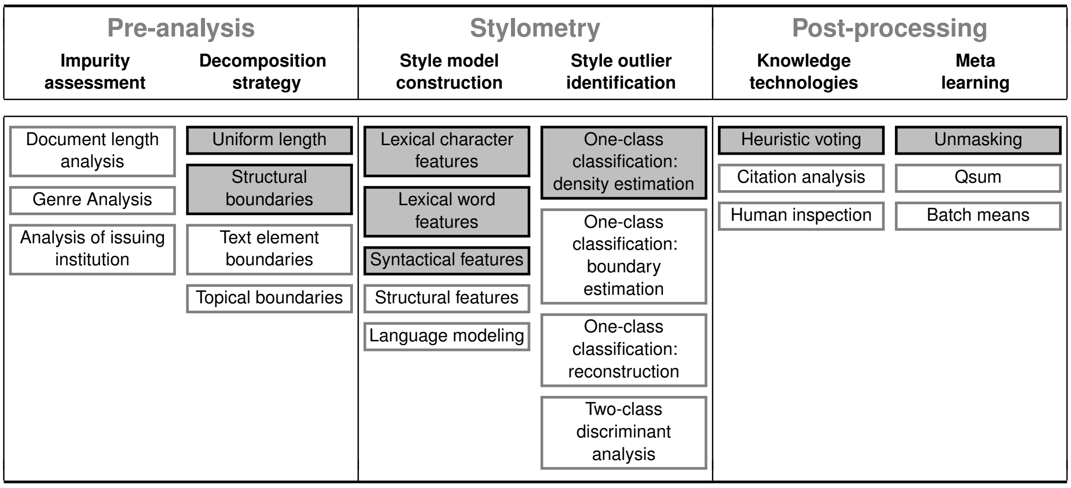
Table 4.13 organizes the building blocks to operationalize these stages. Each column lists methods that can be applied, combined, or adapted in order to address a certain subtask in the entire authorship verification process. If this happens in a skillful manner we may end up with an analysis process comparable with the power of human readers, however, their salient strength is the integration of context-dependent meta-knowledge in the analysis. The following subsection discusses the stylometry building block. For a discussion of all three stages in greater detail see Stein, Lipka, and Prettenhofer [294].

4.6.3 Stylometry

Style model construction The statistical analysis of literary style is known as stylometry, and the first ideas date back to 1851 [119]. The automation of this task requires a quantifiable style model. Efforts in this direction became a more active research field in the 1930s [358, 349, 86]. Meanwhile various stylometric features, also termed style markers, were proposed. They measure writer-specific aspects, such as the vocabulary richness [121, 349], the text complexity [86], or reader-specific grading levels that are required to understand a text [64, 147, 52]. However, the mentioned style markers were developed to judge longer texts ranging from a few pages up to book size.

The style model construction has to consider the decomposition strategy: features have various strengths and pose various constraints on text length, genre, or topic variation. Since text plagiarism typically relates to sections that are

Table 4.13: *Building blocks of an intrinsic plagiarism analysis. The first two columns list pre-analysis methods, the third and the fourth column list the modeling and classification methods, which form the heart of the intrinsic plagiarism analysis, and the last two columns list post-processing methods for improving the analysis quality. The highlighted blocks indicate the employed technology for the analysis in this section. From Stein, Lipka, and Prettenhofer [294].*



Pre-analysis		Stylometry		Post-processing	
Impurity assessment	Decomposition strategy	Style model construction	Style outlier identification	Knowledge technologies	Meta learning
Document length analysis	Uniform length	Lexical character features	One-class classification: density estimation	Heuristic voting	Unmasking
Genre Analysis	Structural boundaries	Lexical word features	One-class classification: boundary estimation	Citation analysis	Qsum
Analysis of issuing institution	Text element boundaries	Syntactical features	One-class classification: reconstruction	Human inspection	Batch means
	Topical boundaries	Structural features	Two-class discriminant analysis		
		Language modeling			

shorter than a single page [193], the decomposition of a document into sections $s_1, \ldots, s_n$ must not be too coarse; it is questionable which of the features work for short sections. It should be clear that features that employ measures such as the average paragraph length are not reliable in general. Meyer zu Eißen et al. [203] investigate the robustness of the vocabulary richness measures Yule's K, Honore's R, and the average word-frequency class. They observe that the average word-frequency class could be called robust: it provides reliable results even for short sections, which can be explained with its word-based granularity.

Table 4.14 compiles an overview of important stylometric features that were proposed so far; we distinguish between lexical features (character-based and word-based), syntactic features, and structural features. Our overview is restricted to the well-known style markers; the features marked with an asterisk were reported to be particularly discriminative for authorship analysis and are used within our stylometric analysis.

Style outlier identification The decomposition of a document d yields a sequence of sections, $s_1, \ldots, s_n$, and the application of a style model yields for these sections a sequence of feature vectors $\mathbf{s}_1, \ldots, \mathbf{s}_n$, which in turn are analyzed with respect to outliers. The identification of outliers among the $\mathbf{s}_i$ has to be solved solely on the basis of target examples and therefore poses a one-class classification problem. Usually, a tailored one-class classification approach should be applied; according to Tax [303] such approaches fall into one of the following three classes:

Density methods Density methods directly estimate the probability distributions

Table 4.14: *A compilation of important and well-known features used within a stylometric analysis. Those implemented within our style model are marked with an asterisk.*

Stylometric feature		**Reference**
Lexical features, character-based		
	Character frequency	[353]
*	Character n-gram frequency/ratio	[149, 256, 136, 157]
	Frequency of special characters (’(’, ’&’, ’/’, etc.)	[353]
	Compression rate	[281]
Lexical features, word-based		
*	Average word length	[119, 353]
	Average sentence length	[119, 353]
*	Average number of syllables per word	[119]
	Word frequency	[213, 119, 157]
	Word n-grams frequency/ratio	[256]
	Number of hapax legomena	[310, 353]
	Number of hapax dislegomena	[310, 353]
	Dale-Chall index	[64, 52]
*	Flesch Kincaid grade level	[86, 147]
*	Gunning Fog index	[101]
*	Honore’s R measure	[121, 310, 353]
	Sichel’s S measure	[310, 353]
*	Yule’s K measure	[349, 119, 310, 353]
	Type-token ratio	[349, 119, 353]
*	Average word-frequency class	[201]
Syntactic features		
	Part of speech	[281, 157]
*	Part-of-speech n-gram frequency/ratio	[152, 157]
*	Frequency of function words	[213, 119, 13, 152, 353, 157]
	Frequency of punctuations	[353]
Structural features		
	Average paragraph length	[353]
	Indentation	[353]
	Use of greetings & farewells	[353, 281]
	Use of signatures	[353, 281]

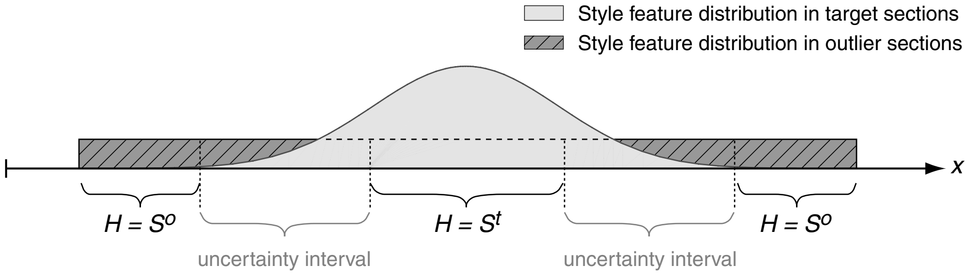

Style feature distribution in target sections
Style feature distribution in outlier sections
x
H = S^o
H = S^t
H = S^o
uncertainty interval
uncertainty interval

Figure 4.9: *Targets and outliers can be separated if they are differently distributed. From Stein, Lipka, and Prettenhofer [294].*

of features for the target class. Outliers are assumed to be uniformly distributed, and Bayes' rule can be applied for separating outliers from the target class (see Figure 4.9 and the following paragraph).

Boundary methods Boundary methods avoid estimating the multi-dimensional density function and focus on the definition of a boundary around the set of target objects. The computation of the boundary is based on the distances between the objects in the target set.

Reconstruction Methods If we are given both the style model representation **s** and the original section s, we may be able to reconstruct **s** from s and to measure the reconstruction error. It is assumed that α captures the domain theory underlying the target class, and the smaller the reconstruction error is the more likely s belongs to the target class.

Tax [303] investigated different representatives for these approaches: mixture of Gaussians, Parzen density, k-center, nearest neighbor, support vector data description, k-means, and self organizing maps. In particular, Tax provides meta-knowledge for selecting among these classifiers by interpreting the presence of outliers, the scaling sensitivity, the number of free parameters, or the sample size.

Stein, Lipka, and Prettenhofer [294] propose the following density method for finding writing-style outliers: let S^t denote the event that a section $s \in \{s_1, \dots, s_n\}$ belongs to the target group (= not plagiarized); likewise, let S^o denote the event that s belongs to the outlier group (= plagiarized). Given a document d and a single style marker x, the maximum a-posteriori hypothesis $H \in \{S^t, S^o\}$ can be determined with Bayes' rule:

$$H = \underset{S \in \{S^t, S^o\}}{\operatorname{argmax}} \frac{\mathrm{P}(x(s) \mid S) \cdot \mathrm{P}(S)}{\mathrm{P}(x(s))}, \tag{4.1}$$

where $x(s)$ denotes the style marker value for section s, and $\mathrm{P}(x(s) \mid S^t)$ and $\mathrm{P}(x(s) \mid S^o)$ denote the respective conditional probabilities that $x(s)$ is observed under a Gaussian and a uniform distribution. The expectation and the variance for x are estimated from $x(s_1), \dots, x(s_n)$. Multiple style markers $x_1, \dots, x_m$ require the accounting of multiple conditional probabilities. Under the conditional independence assumption the naïve Bayes approach can be applied; the accepted a-posteriori hypothesis then computes as

$$H = \underset{S \in \{S^o, S^t\}}{\operatorname{argmax}} \, \mathrm{P}(S) \cdot \prod_{i=1}^{m} \mathrm{P}(x_i(s) \mid S). \tag{4.2}$$

An alternative and, depending on the training corpus, a more powerful approach is the construction of a Gaussian mixture for the $x_1, \ldots, x_m$. The respective weights can be estimated by the linear model of a discriminant analysis.

Coupling stylometry with unmasking In principle, unmasking could be applied to some decomposition $s_1, \ldots, s_n$ of d, assuming an unknown authorship for an s_i, and authorship A for the remaining $d \setminus \{s_i\}$. In most cases, a single section s_i will be too small to be sampled for the unmasking procedure. In this sense the style outlier analysis is a heuristic generator function that helps to construct a potentially plagiarized and sufficiently large auxiliary document of foreign authorships. The underlying search space is the set of all subsets of a document d. Let k, $k < n$, denote the minimum number of sections that must be chosen from a decomposition $s_1, \ldots, s_n$ of d in order to construct an auxiliary document of foreign authorships. With θ as the plagiarized portion of d, $k' = \lceil \theta \cdot n \rceil$ defines an upper bound for the number of sections that can be plagiarized at all. Hence, a brute-force analysis of d has to investigate r auxiliary documents, with

$$r = \binom{n}{k} + \ldots + \binom{n}{k'}, \quad k < k'.$$

An unmasking analysis of r document pairs will not be tractable in most cases; the preceding style outlier analysis enables one to concentrate on a very small number of auxiliary documents.

4.6.4 Evaluation

This section reports on the effectiveness of the operationalized analysis chain. To give the reader an idea of the entire process and its neuralgic points, Figure 4.10 illustrates important analysis aspects: the top row shows documents with non-plagiarized sections (light gray), plagiarized sections (dark gray), and sections spotted by the classifier (hashed); the middle row shows the micro and macro-averaged outlier classification effectiveness; the bottom row shows the heuristic voting and unmasking for critical stylometric analysis cases. The remainder of this section gives full particulars.

Corpus To run analyses on a large scale one has to resort to "artificially plagiarized" documents. We use a subset of the corpus that has been constructed for the intrinsic plagiarism analysis task of the PAN'09 competition. The PAN'09 corpus comprises about 3 000 generated cases of intrinsic plagiarism, more

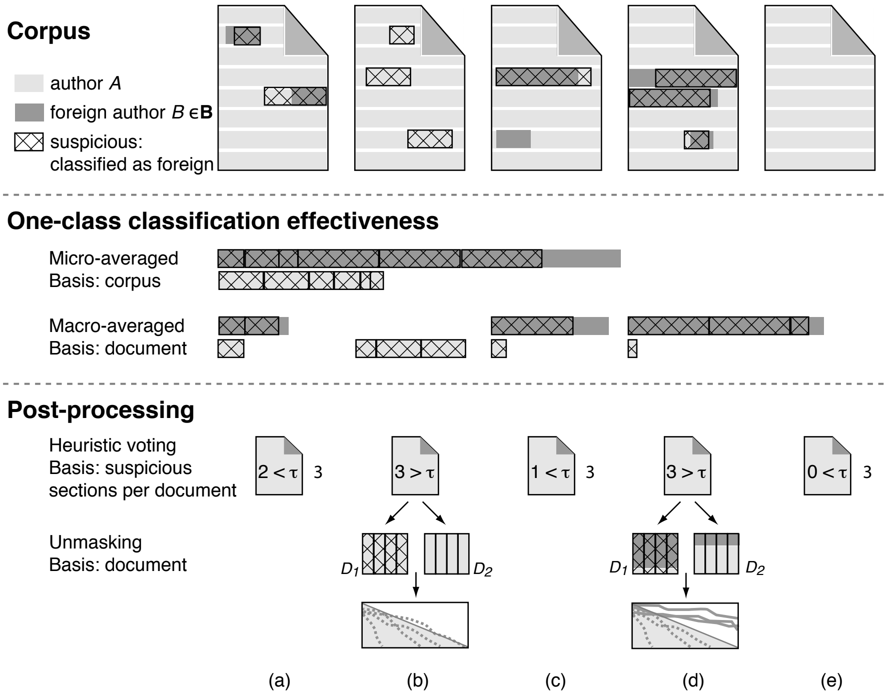


Figure 4.10: *Illustration of important analysis aspects. Corpus: a set of documents from author A, containing sections from a foreign author B ∈* **B***. One-class classification effectiveness: micro- and macro-averaged analysis of the classification effectiveness. Post-processing: heuristic voting decision which may be verified by unmasking. Final classification: true positives (d), true negatives (b, e), false negatives (a, c). From Stein, Lipka, and Prettenhofer [294].*

Table 4.15: *Selected summary statistics of the four test collections. The statistics of the columns 2-4 are per collection and consider both the plagiarized and the non-plagiarized documents; the statistics of the columns 5-7 are per document and consider only the plagiarized documents of a collection.*

Collection	**# Documents**		**# Sections** (total)		**# Sections** (median)		**Impurity** (avg.)
	plag.	non-plag.	plag.	non-plag.		plag.	non-plag.
1	231	231	2 067	44 316	8	76	0.087
2	178	178	451	9 560	2	27	0.090
3	178	178	4 744	21 896	24	50	0.304
4	188	188	1 871	7 814	9	22	0.326

precisely, cases of style contamination with varying degrees of obfuscation. The corpus is based on books from the English part of the Project Gutenberg and contains predominantly narrative text. Sections of varying length, ranging from a few sentences up to many pages, are inserted into other documents according to heuristic placement rules. In addition, a certain obfuscation of the inserted sections is performed by replacing, shuffling, deleting, or adding words.

For our experiments the documents of the PAN'09 corpus are uniformly decomposed into candidate sections of 5 000 characters; each candidate section s in turn is categorized as being either non-plagiarized, if s contains no words from an inserted section, or plagiarized, if s consists of more than 50% inserted sections. Otherwise s is discarded and excluded from further investigations. Documents with less than seven sections are removed from the corpus because they are considered to be too short for a reliable stylometric analysis.

To study the effect of the document length and impurity on the effectiveness of our analysis chain, four disjoint collections are compiled. Two levels of document lengths are introduced (short versus long) and combined with two levels of impurity (light versus strong). Short documents consist of less than 250 000 characters, which corresponds to approximately 40 000 words. The impurity θ of a document is defined as the portion of plagiarized characters, i.e., characters that belong to an inserted section. A document has a light impurity if $\theta \leq 0.15$, and it has a strong impurity if $\theta > 0.15$. Finally, the number of plagiarized documents per collection is set to 50%. The resulting test collections exhibit varying degrees of difficulty, both in terms of training sample scarcity (document length) and class imbalance (impurity). We number the collections according to their level of difficulty and show selected summary statistics in Table 4.15.

Evaluation of stylometry The style outlier identification is approached as a one-class classification problem; in particular, the density estimation method

[7] The word "worst" for the POS tri-gram features designates the worst under the 5 best tri-grams.

Table 4.16: *Stylometric features ranked by their isolated F-measure effectiveness within a style outlier detection task.*

Stylometric feature	***F*-measure**	
	best	worst
Average number of syllables per word	0.733	0.534
Gunning Fog index	0.726	0.452
Flesch Reading Ease Score	0.721	0.466
Frequency of the word: of	0.701	0.425
Average word length	0.700	0.355
Honore's R measure	0.696	0.394
Flesch Kincaid grade level	0.690	0.453
Frequency of the word: the	0.663	0.334
Yule's K measure	0.653	0.285
Part-of-speech trigrams[7]	0.630	0.290
Average word-frequency class	0.601	0.339
Frequency of the word: which	0.587	0.093
Frequency of the word: or	0.578	0.100
Consonant-Vowel-Consonant tri-gram	0.571	0.337
Frequency of the word: the	0.560	0.336
Frequency of the word: and	0.548	0.317
Frequency of the word: by	0.542	0.173
Vowel-Consonant-Vowel tri-gram	0.527	0.301
Frequency of the word: i	0.503	0.157
Frequency of the word: so	0.490	0.066
Frequency of the word: a	0.486	0.156
Frequency of the word: that	0.481	0.177
Frequency of the word: they	0.471	0.047
Frequency of the word: on	0.469	0.141
Frequency of the word: not	0.467	0.143
Frequency of the word: was	0.460	0.190
Vowel-Vowel-Consonant tri-gram	0.446	0.201
Frequency of the word: but	0.445	0.120

Table 4.17: *Effectiveness of the one-class classifier in the stylometric analysis stage. The target class contains all sections from the original author A; the outlier class contains the sections of a foreign author $\neq A$.*

Collection	**Target class**			**Outlier class**		
	Prec	*Rec*	*F*	*Prec*	*Rec*	*F*
1	0.98	0.91	0.94	0.20	0.52	0.29
2	0.89	0.90	0.89	0.34	0.32	0.33
3	0.98	0.64	0.77	0.10	0.78	0.18
4	0.89	0.64	0.74	0.27	0.64	0.38

as described in Section 4.6.3 is applied to identify spurious sections in a document. To capture the idiosyncratic writing style of an author a diverse set of style markers is employed: lexical character features, lexical word features, and syntactical features. Among the employed style markers are the classical measures for vocabulary richness, text complexity, as well as style markers that have been reported to be particularly discriminative for authorship analysis, such as character n-grams and the frequency of function words. To capture syntactic variations in writing-style, part-of-speech information in the form of part-of-speech trigrams is exploited; the tagging is done with the probabilistic part-of-speech tagger QTAG. Table 4.14 summarizes the implemented style markers.

From the large number of several thousand style markers the top k discriminatory style markers are chosen. For this purpose the effectiveness of each style marker within a style outlier detection task is assessed under the univariate model (Equation 4.1). Table 4.16 shows the top 30 style markers in terms of the F-measure concerning the outlier class.

Based on this set of stylometric features a multivariate classifier according to Equation 4.2 is constructed. To reduce the influence of numerical and rounding errors, we resort to the logarithmic variant of Equation 4.2 when computing the maximum a-posteriori hypothesis. Table 4.17 summarizes the achieved classification results for both the outlier class and the target class.

Evaluation of meta-classification To assess the effectiveness of the unmasking approach we evaluate the meta-learner of the basic unmasking procedure. Unmasking is parameterized as follows: documents are represented under the bag-of-words model, defined by the 500 most frequent words (including stop words) of the input document sets D_1 and D_2, without applying stemming or feature selection. The imbalance of the input sets is corrected by over-sampling the minority class, i.e., the outlier class, with the SMOTE approach [55]. In each iteration i of 30 unmasking iterations the best 10 features ranked by the information gain heuristic are removed, and the classification accuracy, Acc_i, of a linear SVM is computed, applying a 5-fold cross-validation. A similar

Table 4.18: *Evaluation of the unmasking meta-learner. Setting: 10-fold cross validated with 100 plagiarized documents and 100 non-plagiarized documents from Collection 2.*

Collection	Non-plagiarized documents			Plagiarized documents		
	Prec	*Rec*	*F*	*Prec*	*Rec*	*F*
1	0.78	0.86	0.82	0.82	0.73	0.77
2	0.77	0.88	0.82	0.48	0.30	0.37
3	0.95	0.94	0.95	0.94	0.95	0.95
4	0.70	0.69	0.70	0.68	0.70	0.69

meta-learner as discussed in Section 4.5 is employed; Table 4.18 reports on its effectiveness.

Originally, the unmasking approach of Koppel and Schler [153] decides for two sets of documents whether or not all documents stem from a single author. If both sets belong to the same author the associated unmasking curve drops (recall the dashed lines in Figure 4.8). This fact is exploited within our analysis chain as we utilize unmasking to filter out cases of alleged plagiarism, which occur because of the insufficient precision in the stylometric analysis stage. In this sense, 1 minus the recall of the non-plagiarized documents defines an upper bound for the false positives rate (see the third row in Table 4.18).

Evaluation of intrinsic plagiarism analysis We evaluate three strategies, from naive to sophisticated, for intrinsic plagiarism analysis for a document d. Under the minimum risk strategy, d is classified as plagiarized if at least one style outlier has been announced for d. Under the heuristic voting strategy d is classified as plagiarized if the detected fraction of outlier text is above a threshold τ. Under the unmasking strategy d is classified as plagiarized if the detected fraction of outlier text is above a threshold $\tau_{\neq}$; d is classified as non-plagiarized if the detected fraction of outlier text is below a threshold $\tau_{=}$; for all other cases unmasking is applied. Note that the values for τ, $\tau_{\neq}$, and $\tau_{=}$ are collection-dependent. Table 4.19 summarize the results.

4.6.5 Summary

Intrinsic plagiarism analysis is the spotting of sections with undeclared writing-style changes in a text document. Intrinsic plagiarism detection is a difficult one-class classification problem that cannot be tackled with a single technique and requires the combination of sophisticated algorithmic and statistical building blocks.

We report on the effectiveness of stylometry, whereas finding elements of the target class (non-plagiarized sections) can be done with a high precision and recall. This is not possible for elements of the outlier class (plagiarized section). We show how the outcome of a style outlier identification can be improved by

Table 4.19: *Overall effectiveness of different solution strategies: minimum risk (column 2-4), heuristic voting (column 5-7), and unmasking (column 8-10).*

Collection	Minimum risk			Heuristic voting				Unmasking			
	Prec	*Rec*	*F*	τ	*Prec*	*Rec*	*F*	$[\tau_{=}; \tau_{\neq}]$	*Prec*	*Rec*	*F*
1	0.50	1.00	0.66	0.1	0.55	0.57	0.63	[0.1; 0.5]	0.83	0.50	0.62
2	0.50	1.00	0.66	0.1	0.50	1.00	0.66	[0.1; 0.5]	0.66	0.57	0.67
3	0.50	1.00	0.66	0.2	0.69	0.30	0.42	[0.2; 0.8]	0.72	0.30	0.43
4	0.50	1.00	0.66	0.2	0.52	0.97	0.68	[0.2; 0.8]	0.98	0.60	0.74

unmasking, which is analyzed in Section 4.5, for the more general question "Does a document contain a plagiarized section?".

4.7 Bibliography

Bag of phrases such as word sequences (n-grams) [208, 265] and bag of words are the most referred type of representations in text classification. These representations are usually comprised of words as features, whereas stop word removal, frequent word removal, stemming, as well as normalization and standardization are common preprocessing routines.

Besides the development of features specialized for (non-standard) text classification, feature selection, extraction, and generation are important related research fields, which are more general and not always restricted to text representations.

Feature selection Feature selection is the field of reducing the complexity of a given representation by selecting reliable features that form the basis of further text representations. Basic feature selection methods remove zero-variance and redundant features by examining feature characteristics on a training sample [211]. Advanced feature selection methods rely on the evaluation of the discriminative power of a feature or a subset of features. The interplay of features in a subset influences the effectiveness of the classifier, which makes the selection of the most effective subset NP-hard.

Wrappers evaluate the discriminative power by training a chosen learning algorithm and estimate its effectiveness with a standard evaluation approach and a chosen effectiveness measure [135, 151]. Wrappers are universal but computationally expensive and tend to overfitting. For selecting a subset of feature, exhaustive, greedy, or heuristic search algorithms can be utilized. Backward selection was applied for the first time by Marill [194], genetic algorithms by Vafai and De Jong [312], and branch and bound methods by [214]. Branch and bound is exclusively applicable under the assumption that a larger feature subset can only increase the effectiveness, which is rarely the case. Measures for the

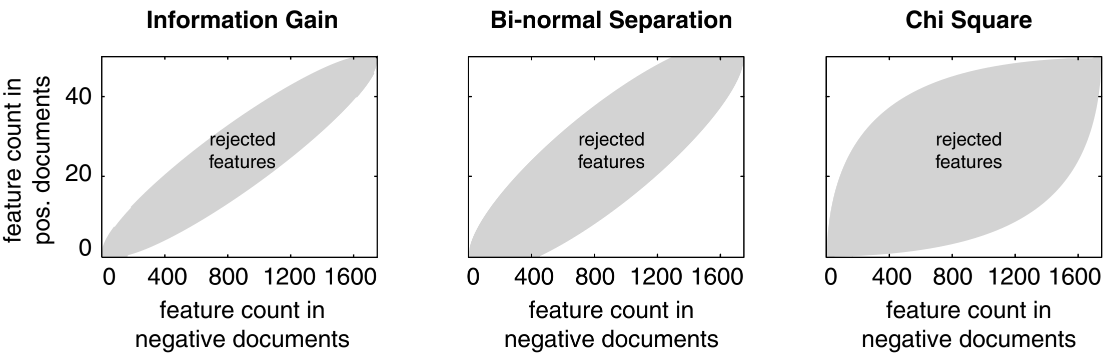


Figure 4.11: *The characteristics of rejected features when the 100 most informative features are selected by the scores information gain, chi-squared, and bi-normal separation. Features that occur often in one class but rarely in the other class are discriminative. Stop words are typically located on the diagonal. This illustration is based on Figure 1 in [87].*

effectiveness evaluation such as mean squared errors or correlation coefficients were proposed in this context by Mallows [191], first.

Another way of feature selection is to compute for each feature a score, which is interpreted as their discriminative power, and to keep the top scoring features. Common scores are information gain, mutual information, chi-squared, and bi-normal separation [3]. Based on [87], Figure 4.11 illustrates their effects in text classification. It should be noted that all of these scores evaluate the document frequencies of the features in isolation. Therefore, the main disadvantage is that resulting effects from the combination of features are not considered. For further methods refer to the work by Guyon and Elisseeff [103] and Liu and Motoda [184]; for further scores refer to the comparative study by Yang and Pedersen [344].

Feature extraction Feature extraction is the field of transforming a given representation by extracting reliable features that form the basis for further text representations. The majority of feature extraction methods is data-driven and combines existing features by means of multivariate analyses. The most important methods are: linear discriminant analysis [3], principal component analysis [226], correspondence analysis [115], factor analysis [279, 286], singular value decomposition [297, 350], clustering algorithms [169, 17, 278, 19, 2], nonlinear dimensionality reduction algorithms [166] (especially maximum variance unfolding [329]), multilinear subspace learning algorithms [190], and multidimensional scaling algorithms [61].

Feature generation Feature generation is the field of generating or constructing reliable features that form the basis for further text representations. Information extraction methods "augment" text representations, for example by part-of-speech tags or linguistic patterns [248], or by named entities [160].

Also, WordNet synonym set identifiers can be used to represent words with equivalent semantics [67, 98, 82], which leads to a more abstract representation overall. Using Wikipedia as external knowledge, texts can be represented by their similarities to Wikipedia articles, which function as concepts, cf. explicit semantic analysis by Gabrilovich and Markovitch [92]. More dictionary-based methods are found, for example, in [311, 323]. One advantage is that infrequent words, which are usually ignored, are still captured by the external knowledge. Furthermore, with the exploration of the multilingualism in Wikipedia, cross-lingual representations can be compiled [20]. Texts are represented by their similarities to a reference set of Wikipedia articles, cf. cross-lingual explicit semantic analysis by Potthast et al. [234] and Anderka, Lipka, and Stein [6]. More cross-lingual representations are cross-lingual structural correspondence learning by Prettenhofer and Stein [236] (based on [27]).

Chapter 5

Towards effective text classification in the wild

Employing a carefully engineered classification solution in the wild, i.e., under non-laboratory conditions, often leads to dissatisfaction, even if it has a reasonable effectiveness under laboratory conditions—what are the roots? We consider the relationship between the populations and samples to be decisive. If the ratio between population and sample sizes becomes extreme, the classifier and its evaluation are misled, as the sample is no longer representative. This is also the case if an independently and identically distributed sample contains almost exclusively examples from the majority class, which often happens if the population is highly imbalanced. If the degree of imbalance is unknown, the outcome of an evaluation is difficult to interpret. Even worse, a good deal of information filtering tasks have dynamic populations and noisy class labels, which makes it impossible to apply standard machine learning research and developments.

We face these problems of representativeness in multiple ways, depending on the particular conditions:

Small sample but balanced classes In Section 5.1, we propose robust feature engineering as a suitable modeling principle if the samples are too small in relation to the population. The idea is to control the generalization capabilities of a classification solution via the inductive bias that is introduced by coarse text representations, cf. 3.1.

Imbalanced but stable classes In order to sample more examples from the minority class in imbalanced problems, we propose an active learning strategy in Section 5.2, which is trained on past classification tasks. Active learning has the goal of choosing a small, balanced sample of informative examples, which replaces a larger independently and identically drawn sample.

Dynamic or noisy complementary but stable and noiseless target classes Modeling a classification task as a one-class classification problem has the advantage that the training process is not affected by class imbalances and dynamic or noisy complementary classes. This is motivated by detecting information quality flaws in Wikipedia articles, whereas examples of flawed articles belong to a well-defined concept. In contrast, non-flawed articles belong to an unspecified and unknown concept. Section 5.3 and 5.4 concentrate on learning from the target class only and on improving one-class classifiers; we propose an ensemble method, which combines several one-class classifiers.

5.1 Improving generalization capabilities via robust feature engineering

Almost all statistical models underestimate the likelihood of unseen events, which is a severe problem for machine learning. In addition, training examples are often hard to acquire, and the smaller the training sample, the more unseen events exist. Furthermore, if the feature space has a high dimensionality, the number of possible events grows. This is also known as the "high dimensionality, small sample size" problem.

The existing research addressing this problem can be distinguished into the following areas: (1) theoretical analysis of sample complexity, (2) multiple evaluations of a training sample **S**, and (3) semi-supervised learning.

(1) The sample complexity is related to the question of how many training examples are needed such that a learner converges with high probability to a successful hypothesis [206]. A key factor is the size of the learner's underlying hypothesis space. There are upper bounds linear in $VC(H)$, the Vapnik-Chernovenkis dimension of the hypothesis space, and logarithmically in $|H|$, the size of the hypothesis space [30, 318].

(2) A multiple evaluation of training samples can be realized with ensemble classifiers and collaborative filtering techniques [280, 220, 49]. An ensemble can be considered to be a committee of experts, each of which is focusing on different aspects of the training sample, and the combined expertise can alleviate the negative impact of a small sample **S**.

(3) Semi-supervised learning approaches are appropriate, if they are trained on a small training sample **S** and a large representative sample of unlabeled examples [276, 5]. A promising approach in this regard is the integration of domain knowledge into the learning phase [75].

In contrast to the related work, we put the feature engineering in the focus. In particular, we propose to identify the robustness of a model with the inductive bias that is intentionally introduced within the feature engineering.

We provide a case study that allows us to observe these considerations by comparing variants of the vector space model resulting from different feature engineering functions. The studied functions are independent of any utilization of training examples.

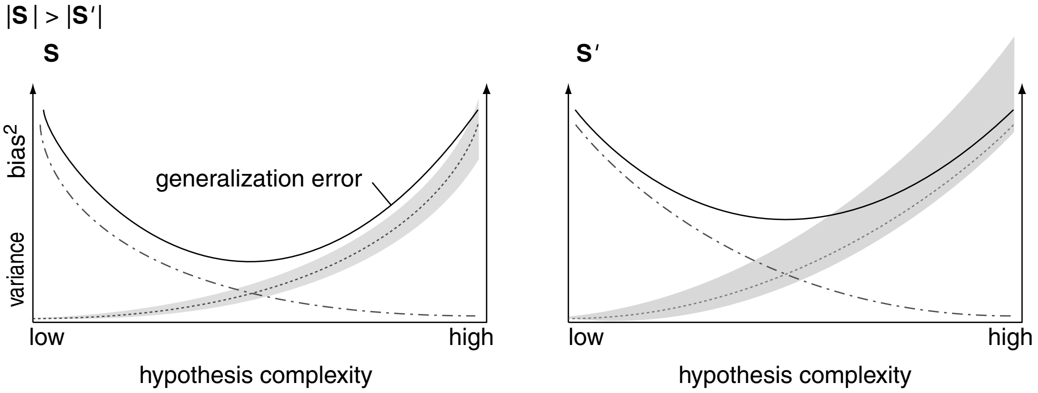


Figure 5.1: *The bias-variance trade-off and the estimation of the generalization error (the sum of variance and bias squared) based on the samples* **S** *and* **S**′ *with* $|\mathbf{S}| > |\mathbf{S}'|$*. If* **S** *is large enough, the estimate of the generalization error is tighter than if a small* **S**′ *is used. For a small* **S**′ *and a complex hypothesis, the generalization error tends to be underestimated because of the variance estimate. (1) The risk of the variance estimator grows biquadratically with the hypothesis complexity; in contrast, the risk of the bias estimator only grows quadratically, cf. [301]. (2) With decreasing the size of the sample, the difference between the risks of these estimators grows exponentially.*

The idea of robust models, proposed in this section, can be regarded as a model selection paradigm that gives preference to the apparently inferior model with a larger model formation bias. We regard, however, the automatic construction of robust models as an important open problem in machine learning.

5.1.1 Robust Models

The investigations of this section are motivated by the extreme relations in information filtering. We are working on classification tasks such as Web genre analysis or the semi-automatic maintenance of large repositories, where the size ratio v between the sample **S** (comprising training and test data) and the set of unseen documents is close to zero. As a consequence, even sophisticated learning strategies are misguided by **S** if the feature vectors $\mathbf{d} \in \mathbf{S}$ consist of many and highly variant features. The reason for the misguidance is that the concept of representativeness inevitably gets lost for $v \ll 1$ and, as a result, it is no longer possible to apply a standard model or feature selection.

We argue that even in such extreme learning situations, classifiers can be built that generalize well: the basic idea is to withhold information contained in **S** from the learner. Conceptually, such a restriction cannot be left to the learner but must happen intentionally, by means of a task-oriented feature engineering *by the engineer*. The statistical feature selection strategies reviewed in Section 4.7, cannot address the lack of training samples. These strategies can be exploited if the number of features is high and the training sample is both plentiful and representative. Only then it can be utilized for building a classifier with a less

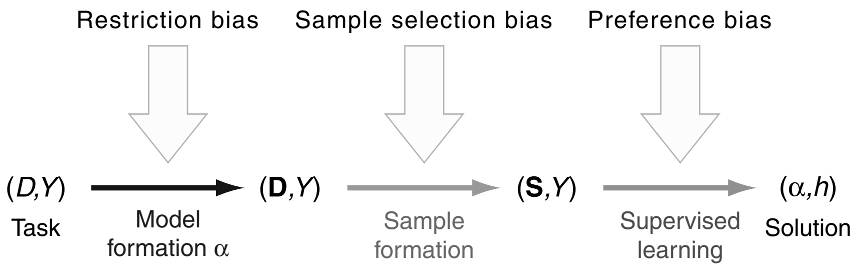


Figure 5.2: *Illustration of a classification task (D, Y) and its solution. The feature engineering function α associates real-world objects with feature vectors. A restriction bias is introduced by the feature engineering; other biases are introduced within the subsequent steps.*

complex hypothesis structure and an improved generalization characteristic [103, 339].

Again, the predictive behavior of a classifier is rooted in its inductive bias as described in Section 3.1). Biases are often implicitly introduced by the employed text classification model, that is, by feature engineering, by sampling, and by the learning algorithm. Given a classifier in a concrete learning situation, the statistical bias quantifies the error that is caused by this simplification, while the inductive bias can be regarded as the rationale (the logical argument) for this error. Accepting a higher statistical bias will reduce the variance of the learned classifier and may entail a lower generalization error—a connection which is known as bias-variance trade-off as depicted in Figure 5.1.

If only a very small number of training examples is available, choosing among different complex models by determining the best bias-variance trade-off becomes a game of chance. All learning methods that try to build classifier with a minimum generalization error, rely on the assumption that the examples are representative.

Recall building a classification solution (Figure 5.2). The starting point is a classification task (D, Y), where we are given a set of documents D, the population, which can be classified by a real-world classifier into k classes $Y = \{1, \ldots, k\}$. A real-world classifier should be understood as a decision machine that is unrestricted in every respect. By contrast, computer algorithms work on an abstraction $\mathbf{d}$ of a document d. The process of deriving $\mathbf{d}$ from d is denoted as α, $\alpha : D \rightarrow \mathbf{D}$. $\mathbf{D}$ comprises the feature vectors of the population; it constitutes a multiset, implying the identity $|D| = |\mathbf{D}|$ and preserving in $\mathbf{D}$ the class distribution of D. The task of an inductive learner is to build an approximation h of the target concept c, exploiting only information contained in a sample $\mathbf{S}$ of training examples $\{(\mathbf{d}, c(d))\}$. The hypothesis h is characterized by its generalization error, $err(h)$, which is the probability of wrong classification:

$$\mathrm{P}(h(\mathbf{d}) \neq c(d)).$$

The test error $err_{\mathbf{S}}(h)$, which is measured on the sample $\mathbf{S}$, is an estimator of $err(h)$. The learning algorithm selects a hypothesis h from the space H

Table 5.1: *Document distribution in the top-level categories of RCV1.*

Top-level category	Number of documents
corporate/industrial	292 348
economics	51 148
government/social	161 523
markets	158 749

of possible hypotheses, and hence H defines a lower bound for $err(h)$. This lower bound is denoted as structural bias $err(h^*)$ and quantifies the expected difference between an optimum hypothesis $h^* \in H$ and the target concept c.

Choosing between different functions $\alpha_1, \ldots, \alpha_m$ implies choosing between different text representations $\mathbf{D}_1, \ldots, \mathbf{D}_m$ along with different hypotheses spaces $H_{\alpha_1}, \ldots, H_{\alpha_m}$, and hence to introduce a more or less rigorous structural bias. If the training sample is plentiful, the best model can be found by minimizing $err_{\mathbf{S}}(h)$ against the different representations; if the training sample is scarce, we even may prefer α_i over α_j although the former is outperformed under $\mathbf{S}$:

$$err_{\mathbf{S}}(h^*_{\alpha_i}) \; > \; err_{\mathbf{S}}(h^*_{\alpha_j}),$$

where $h^*_{\alpha_i} \in H_{\alpha_i}$, $h^*_{\alpha_j} \in H_{\alpha_j}$, and $i \neq j$. We introduce a higher restriction bias than suggested by $\mathbf{S}$, accepting a higher error $err_{\mathbf{S}}$, but still expecting a lower generalization error:

$$err(h^*_{\alpha_i}) \; < \; err(h^*_{\alpha_j})$$

We declare the model under α_i to be more robust than the model under α_j or to be a robust model for the task (D, Y).

5.1.2 Case study: Topic categorization

The following experiments evaluate the behavior of the generalization error err, the sample error $err_{\mathbf{S}}$, and the relation between err and $err_{\mathbf{S}}$. In our study we vary vector space representations by employing different functions α while keeping the learning algorithm unchanged. This way, the difference in the classification model's robustness is reflected by the classification effectiveness of the obtained solutions. The learner in the setting is again a linear SVM and (D, Y) is a text categorization task on the Reuters Corpus Volume RCV1 [171]. We consider the corpus in its entirety in the role of the population D. The set Y of class labels is defined by the four top-level categories in RCV1: corporate/industrial, economics, government/social, and markets. The corpus contains $|D| = 663\,768$ uniquely classified documents whose distribution is shown in Table 5.1.

The functions α_i lead to different text representations $\mathbf{D}_i$. Let $\mathbf{S}$ be a sample, drawn independently and identically distributed from $\mathbf{D}_i$, with $|\mathbf{S}| = 800$. The extreme ratio of $v = 0.0012$ between the sizes of $\mathbf{S}$ and $\mathbf{D}_i$ reflects a

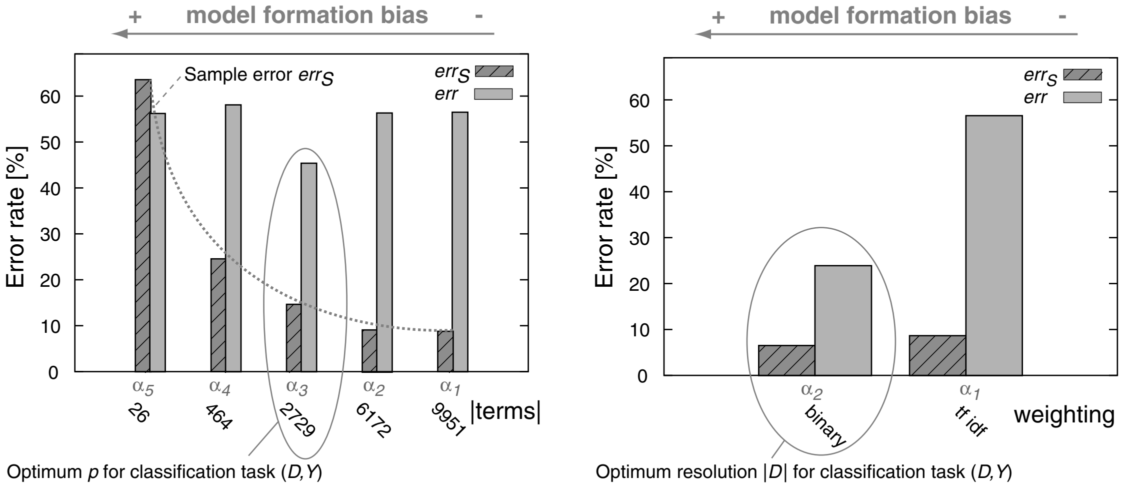


Figure 5.3: *Cross-validated error estimates (hashed bars) and generalization errors (plain bars) for a linear SVM trained on 800 examples.* **(left)** *Five different solutions (α_i, h) of (D, Y). The $\alpha_1, \ldots, \alpha_5$ affect H by the employed number of index terms.* **(right)** *Two different solutions (α_i, h) of (D, Y). α_1 and α_2 affect H by using a different granularity for the feature variable domains.*

typical information retrieval situation as it is encountered in the real world; in fact, $v = 0.0012$ may still be considered optimistic.

It is important to note that the introduced bias during the experiments comes from uninformed feature engineering strategies. We do not employ any statistical feature selection, extraction, or generation methods. Furthermore, the mass of words that are captured by each representation remains the same, i.e., a representation becomes smaller by introducing a more general representation. In contrast, feature selection might remove features and, therefore, the respective words are ignored in the on-going processing of examples.

Experiment 1 For a document $d \in D$, $\alpha_i(d)$ computes a vector space model, where $i = 1, \ldots, 5$, is associated with a certain number p of used index terms (see the x-axis in Figure 5.3 (left) for the actually chosen values for p). The reduction of the feature number p is achieved by introducing prefix equivalence classes for the index terms: the weights of words that start with the same letter sequence are added to build the weight of the new feature that represents the equivalence class. In our experiments the prefix length varies between 1 and 10.

The plot in Figure 5.3 (left) reveals, as expected, that the cross-validated error estimates (hashed bars) increase with the impairment of the vector space model. Interestingly, this monotonic behavior cannot be observed for the generalization error: for $p = 2\,729$ the value becomes minimal, a further reduction of p leads to underfitting. To understand the importance of this result, recall that the generalization error cannot be observed in information retrieval practice. Put another way, the best solution for (D, Y) can be missed easily since only the analysis results with respect to $\mathbf{S}$ are at our disposal.

Experiment 2 We now modify α_i by coarsening the feature domain D of the index terms, going from the *tf·idf* model to the boolean model. Figure 5.3 (right) shows the results for the two extremal α_i. The cross validated errors for both models are pretty close to each other; in fact, they differ only by one percent. Hence, there is a high risk of selecting the wrong model. This is particularly crucial here since the difference between the achievable generalization errors is enormous.

That the err_S statistic may lead one astray, even if it relies on cross-validation, has been observed and discussed before [260]. Our analyses go beyond these and similar results: Firstly, we report on realistic information retrieval experiments and the current practice of experiment implementation and experiment evaluation. Secondly, and presumably more important, the focus of our analyses is on the impact of α. The above analysis is distantly related to the feature selection problem, which also can cause some bias on the estimates of classifier parameters. This kind of bias is also known as "feature subset selection bias" or simply "selection bias" [277].

5.1.3 Summary

Robust models are a means for reducing the overfitting problem for classification tasks where the ratio between the training sample and the population of unseen examples is extremely small. We argue to identify the bias that is introduced within the feature engineering with the robustness of the resulting classification solutions. In a case study, we analyze the impact of the model formation bias on the generalization capabilities, and we observe that the idea of robust models is highly usable; it captures effects on the generalization error that cannot be attributed to properties of the inductive learner nor to the hypothesis structure. Nevertheless, the systematic determination of robust models is an open problem in machine learning.

5.2 Sampling informative examples via advanced active learning

Active learning, a sub-discipline of supervised learning, aims to achieve higher accuracy with fewer training examples. This is accomplished by allowing the active learning framework to choose the examples from which it learns. An active learner poses queries, usually in the form of unlabeled examples to be labeled by an oracle. From a statistical point of view, a query strategy builds a sample, which represents the population's characteristics that are informative for building discriminative hypotheses. Active learning is regarded as a sampling strategy β in our context.

Active learning has numerous applications in the fields of information extraction, speech recognition, and filtering. Learning to classify documents (e.g., articles or webpages) requires that users label each document using a given class scheme. Standard query strategies for active learning are based upon various heuristics, for example, uncertainty sampling [170], query by committee [268], expected model change [267], expected error reduction [251], and variance reduction.

Often, uncertainty sampling is the most effective strategy, which selects examples close to the current decision boundary. Especially discriminative classifiers benefit from uncertain examples most as they have the largest effect on the hypothesis determination. Other examples are not required as long as there is no need to model the class conditional probability distribution.

Uncertainty sampling can be too narrow in exploring the population. Therefore, we introduce a new query strategy based upon machine learning. The goal is to predict how informative unlabeled examples are with respect to a given text classification task. We refer to this as "learning to active learn" and to the corresponding query strategy as "predicted meta-sampling".

5.2.1 Learning to active learn

The idea is that a classifier discriminates between examples that are informative or non-informative with respect to a given classification task. This classifier is used to predict examples that should be queried and added to the training sample in an active learning iteration. The predicted meta-sampling strategy uses meta-knowledge, namely, whether an example has been useful in a past learning scenario. We refer to the corresponding classifier as "meta-sampling classifier" h_{meta}.

In what follows, we will introduce the general active learning procedure, describe the process of generating metadata to train h_{meta}, and propose a model for representing metadata.

Active learning For active learning, the examples are split into three subsets, namely the initial training sample $\mathbf{S}_{train}$, the test sample $\mathbf{S}_{test}$, and the unlabeled sample $\mathbf{U}$. $\mathbf{S}_{train}$ is used for building a base classifier h_{base} in the first step of active learning, whereas its weighted average F-measure is evaluated on $\mathbf{S}_{test}$. The unlabeled sample $\mathbf{U}$ is processed by the active learning approaches and used for selecting an example that is queried (labeled by an oracle or an expert) and added to the initial training sample.

We consider the following general active learning approach, which consists of two components, a learning algorithm L (here, linear SVMs) and a query strategy Q:[1]

[1] Note, h is a hypothesis and c the target concept.

Active Learning

```
Input: $L$, $Q$, $\mathbf{S}_{train}$, $\mathbf{S}_{test}$, $\mathbf{U}$
    Init: $i = 1$, init $Q$
    while $i \leq$ query budget do
        select: $\mathbf{d} = Q(\mathbf{U})$
        query $\mathbf{d}$ to obtain $y = c(d)$
        remove: $\mathbf{U} = \mathbf{U} \setminus \mathbf{d}$, add: $\mathbf{S}_{train} = \mathbf{S}_{train} \cup \mathbf{d}$, train: $h_{base} = L(\mathbf{S}_{train})$
        evaluate $h_{base}$ on $\mathbf{S}_{test}$, update $Q$
        $i = i + 1$
    end while
```

Generating metadata Metadata is generated on previous labeled examples, the validation sample, which is gathered from solved classification tasks. The validation sample is split into an initial training sample $\mathbf{S}_{val,train}$, a test sample $\mathbf{S}_{val,test}$, and an unlabeled sample $\mathbf{U}_{val}$. The metadata M is computed as follows:

Computation of metadata M

```
Input: $L$, $\mathbf{S}_{val,train}$, $\mathbf{S}_{val,test}$, $\mathbf{U}_{val}$
    Output: $M$
    Init: $i = 1$, init $Q$
    while $i \leq$ query budget do
        select randomly $\mathbf{d} \in \mathbf{U}$
        query $\mathbf{d}$ to obtain $y = c(d)$
        remove: $\mathbf{U} = \mathbf{U} \setminus \mathbf{d}$, add: $\mathbf{S}_{val,train} = \mathbf{S}_{val,train} \cup \mathbf{d}$, train: $h_{base} = L(\mathbf{S}_{val,train})$
        evaluate $h_{base}$ on $\mathbf{S}_{val,test}$
        if effectiveness increased by at least 1% then
            $M \cup (\mathbf{d}, +)$
        else
            if effectiveness decreased by at least 1% then
                $M \cup (\mathbf{d}, -)$
            end if
        end if
        $i = i + 1$
    end while
    return $M$
```

Informally, a base classifier h_{base} is updated in each iteration, after a randomly sampled example is added to the training sample. Then the weighted average F-measure of h_{base} is estimated on the test sample $\mathbf{S}_{val,test}$. The example is added to the metadata M when there is a substantial change of the effectiveness of h_{base}. Its class label is "informative, $+$" if the estimated weighted average F-measure is increased by at least one percent compared with the previously added example, or the class label is "non-informative, $-$" if the effectiveness decreases by at least one percent. The metadata is gathered from several datasets, then balanced, and used for building the meta-sampling classifier h_{meta}. As the learning algorithm for the base classifier, we apply a linear SVM where the confidence of the

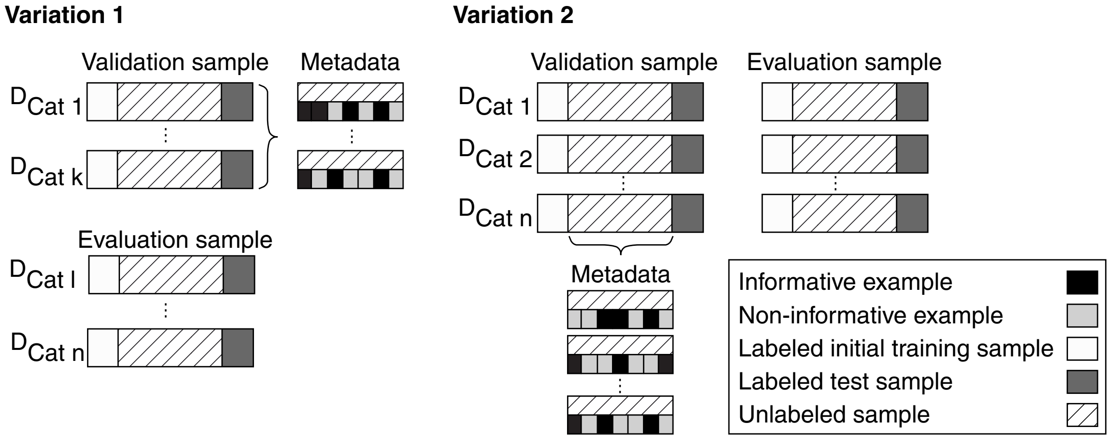


Figure 5.4: *Illustration of the experimental setup. Variation 1: Compute the metadata on a set of Reuters categories and evaluate the query strategies on a separated set. Variation 2: Compute the metadata and evaluate the query strategies on the same set of Reuters categories (validation and evaluation examples are still disjunct).*

prediction is estimated by a logistic regression model [230]. As the learning algorithm for the meta-sampling classifier we apply naïve Bayes [73].

This algorithm for generating metadata is incremental as examples are added while the effectiveness of the base classifier is monitored. In a decremental approach, the interpretation of informativeness is inverse: observing an effectiveness decrease corresponds to an informative example since the removed example has had a positive influence on the classification solution.

Predicted meta-sampling employs the meta-sampling classifier $h_{meta} = L(M)$ by classifying all examples in the unlabeled evaluation sample **U**. Then, examples with the highest predicted informative probability are selected for querying.

Representing metadata The examples used for building the meta-sampling classifier are represented under the following model:

- the disagreement vote using the absolute value of the sum of the predicted classes $\{+1, -1\}$ based on a k-nearest neighbor classifier, a linear SVM, and a naïve Bayes classifier
- the class probability using a k-nearest neighbor classifier (estimated by $1/distance$)
- the class probability estimate using a linear SVM (estimated by logistic regression)
- the class probability estimate using a naïve Bayes classifier

Additionally, we explore descriptive characteristics, such as variance, mean, geometric mean, kurtosis, skewness, maximum value, and minimum value, of the predicted class probability distributions of each classifier, and the distribution of labeled and unlabeled examples.

5.2.2 Evaluation

Corpus We evaluate our proposed learning to active learn approach on the Reuters Corpus Volume I (RCV1) [171], which comprises 806 791 documents, each assigned to one or more categories of an ontology of news articles. In total, 432 499 documents occur in only one path of the ontology. We obtained 38 categories with more than 1 000 documents, whereas each document does not occur in multiple paths of the ontology and is assigned to the deepest category in the path. A Reuters document is represented by its word frequency vector, where the vocabulary consists of 7 392 words that are stemmed and occur at least ten times in the corpus. All digits and special characters are ignored. We construct 38 datasets following the one versus all principle. Each dataset has 1 000 documents of one category and another 1 000 documents sampled from all 37 000 documents belonging to the other categories.

The following sampling strategies (query strategies) are implemented:

Random sampling, β_1 Query randomly chosen examples from **U**.

Uncertainty sampling, β_2 Query the examples from **U** that have the least confident prediction awarded by the base classifier h_{base} [170].

Query by committee, β_3 Query a randomly chosen example from **U** if one classifier in the committee disagrees [268].

Predicted meta-sampling, β_4 Query the most informative example within **U** based on h_{meta}'s predictions.

Gold standard The classifier learns from the training and the unlabeled sample, whereas the labels for the entire unlabeled sample are provided.

In order to compile the metadata M for building the meta-sampling classifier h_{meta}, we separate a sample of validation examples from the Reuters data. We studied two variations as shown in Figure 5.4. In Variation 1, we compute the metadata for the *predicted meta-sampling* query strategy on a set of Reuters categories (validation sample) and apply it on a different set of categories (evaluation sample). In Variation 2, we used the same Reuters categories for metadata computation and active learning evaluation. Each Reuters category is split into validation and evaluation samples.

Experiment: Can the meta-sampling classifier predict informative examples? Variation 1: We randomly select 10 Reuters categories and ran 100 validation iterations, and we use 1% of the given sample as training sample, 69% as unlabeled sample, and 30% as test sample. This setting results in 122 examples for M. The meta-sampling classifier h_{meta} has an accuracy of 54%, a precision value of 0.69, and recall value of 0.15 when finding performance increasing examples.

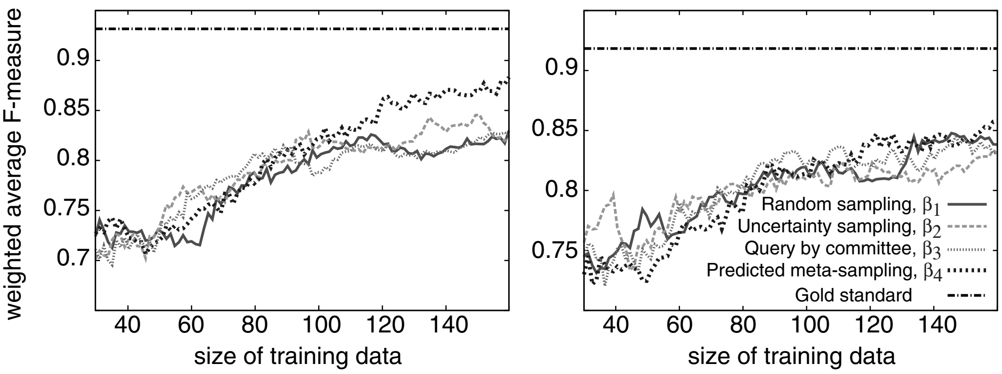


Figure 5.5: *The subfigures show the active learning effectivenesses with different sampling strategies within two Reuters categories out of 18 categories considered in the evaluation. The experimental setup corresponds to Variation 2 in Figure 5.4.*

Variation 2: We randomly select 18 Reuters categories and ran 100 validation iterations, and we use 50% of each category as validation sample and 50% as evaluation sample. Again, the validation and evaluation samples are divided into 1% training sample, 69% unlabeled sample, and 30% test sample. M comprises 228 examples and h_{meta} has an accuracy of 53%, a precision of 0.62, and recall of 0.16 for finding informative examples. In both variations, the precision values of the meta-sampling classifiers imply that informativeness is to some extent predictable.

Experiment: Can active learning benefit from the meta-sampling classifier? Figure 5.5 shows two typical scenarios for experiment Variation 2. The sampling strategy β_1, random sampling, leads in most cases to a stable effectiveness increase when adding examples. The function β_2, uncertainty sampling, is commonly a good choice. The function β_4, predicted meta-sampling, shows a promising effectiveness and is slightly superior compared with the other sampling strategies.

Experiment: Can the meta-sampling classifier be transferred to other domains? Figure 5.6 shows the effectiveness of the active learning approaches in the experiment Variation 1. This setup is apparently harder since the metadata is compiled from completely different Reuters categories. The diagram shows that predicted meta-sampling is performing well over different domains.

5.2.3 Summary

We present an algorithm for learning a sampling strategy, that is, learning to active learn. The evaluation demonstrates that our strategy is effective for some text classification tasks. The prediction of informativeness, which is used for our proposed sampling approach, however, is difficult. The unrepresentative feature

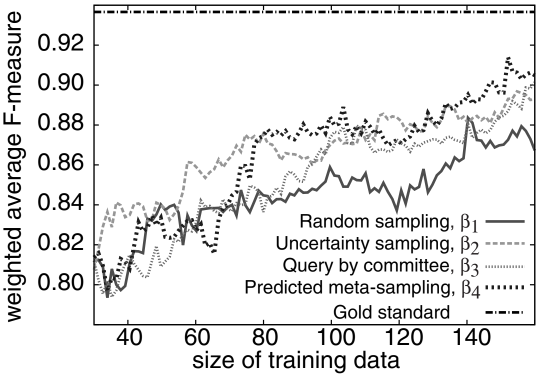


Figure 5.6: *Active learning effectivenesses with different sampling strategies within one RCV1 category. The experimental setup corresponds to Variation 1 in Figure 5.4.*

space and the distribution-dependent informativeness need to be addressed in future work.

5.3 Learning target classes via one-class classification

The majority of text classification tasks suffer from the fact that important characteristics are not represented by the training sample. With a sample that represents at least the target class, a one-class classifier can be trained. As an illustrative example, we argue that the prediction of information quality flaws is essentially a one-class problem. This section gives a formal problem definition of classifying information quality flaws, and it applies a tailored one-class learning approach to address this classification problem. In order to model information quality flaws, we employ the article representation developed by Anderka, Stein, and Lipka [9].

5.3.1 Classifying information quality flaws in Wikipedia

Problem statement Let D be the population of Wikipedia articles and let F be a set of information quality flaws. An article $d \in D$ can contain up to $|F|$ flaws, where, without loss of generality, the flaws in F are regarded as being uncorrelated. A classifier h has to solve the following multi-labeling problem:

$$h : \mathbf{D} \to 2^F,$$

where 2^F denotes the power set of F. Basically, there are two strategies to tackle multi-labeling problems:

multiclass classification a single classifier is learned on the power set of all classes

binary classification for each $f_i \in F$ a classifier $h_i : \mathbf{D} \to \{+1, -1\}$ is learned

Table 5.2: *Two information quality flaws of English Wikipedia articles along with a description and the number of articles that have been tagged. From Anderka, Stein, and Lipka [9].*

Flaw name	Description	Tagged articles
Unreferenced	The article does not cite any references or sources.	273 230
Advert	The article is written like an advertisement.	7 186

Since the high number of classes under a multiclass classification strategy entails a very large number of training examples, the second strategy is favorable.

In most classification problems, training examples are available for all classes that occur at the prediction time, and hence it is appropriate to train a classifier h_i with (positive) examples of the target class f_i and (negative) examples from the classes $F \setminus f_i$. When spotting quality flaws, an unseen article can either belong to the target class f_i or to some unknown class that has been unavailable during training. The standard discrimination-based classification approaches (binary or multiclass) are not applicable to learn a class-separating decision boundary: given a flaw f_i, its target class is formed by those articles that contain flaw f_i, but it is impossible to model the complementary class with articles *not* containing f_i. Even if many counterexamples are available, they could not be exploited to properly characterize the population of the complementary class. As a consequence, we model the classification $h_i(\mathbf{d})$ of an article $d \in D$ with respect to a quality flaw f_i as the following one-class classification problem: Decide whether or not d contains f_i, given a sample of articles containing f_i.

As an illustration, Table 5.2 shows a summary of the flaw "Advert". A large sample of articles that suffer from this flaw can be compiled as 7 186 articles have been tagged with this flaw. Nevertheless, it is impossible to compile a representative sample of articles that have a reasonable writing style and do not advertise. Although many articles are non-flawed featured articles, they cannot be considered a representative sample. Featured articles are a biased group of Wikipedia articles and differ from average articles. Training a binary classifier using featured articles and flawed articles would lead to a biased classifier that is not able to predict flaws on the entire Wikipedia. Also, using random articles and flawed articles to train a binary classier is problematic because random articles are noisy as some of them are not tagged with flaws. Moreover, it is more than likely that the distribution of random articles changes over time.

Method Following Tax [303], three principles to construct a one-class classifier can be distinguished: density estimation methods, boundary methods, and reconstruction methods. Here we resort to a one-class classification approach, which combines density estimation with class probability estimation [111]. There are two reasons for using this approach: (1) Hempstalk et al. [111] show that it is able to outperform state-of-the-art approaches, including a one-class SVM, and (2) it can be used with arbitrary density and class probability estimators. Instead employing an out-of-the-box classifier, we apply dedicated density and class probability estimation techniques to address the problem defined above.

The idea is to use a reference distribution to model the probability $\mathrm{P}(\mathbf{d} \mid f_i')$ of an artificial class f_i', and to generate artificial examples governed by the distribution characteristic of f_i'. For a flaw f_i let $\mathrm{P}(f_i)$ and $\mathrm{P}(f_i \mid \mathbf{d})$ denote the a-priori probability and the class probability function respectively. According to Bayes' theorem the class-conditional probability for f_i is given as follows:

$$\mathrm{P}(\mathbf{d} \mid f_i) = \frac{(1 - \mathrm{P}(f_i)) \cdot \mathrm{P}(f_i \mid \mathbf{d})}{\mathrm{P}(f_i) \cdot (1 - \mathrm{P}(f_i \mid \mathbf{d}))} \mathrm{P}(\mathbf{d} \mid f_i')$$

$\mathrm{P}(f_i \mid \mathbf{d})$ is estimated by a class probability estimator, which is a classifier whose output is interpreted as a probability. Since we are in a one-class situation, we have to rely on the face value of $\mathrm{P}(\mathbf{d} \mid f_i)$. More specifically, $\mathrm{P}(\mathbf{d} \mid f_i)$ cannot be used to determine a maximum a-posterior (MAP) hypothesis among the $f_i \in F$. As a consequence, given $\mathrm{P}(\mathbf{d} \mid f_i) < \tau$ with $\tau = 0.5$, the hypothesis that d suffers from f_i could be rejected. Because of the approximative nature of $\mathrm{P}(f_i \mid \mathbf{d})$ and $\mathrm{P}(f_i)$, the estimation for $\mathrm{P}(\mathbf{d} \mid f_i)$ is not a true probability, and the threshold τ has to be chosen empirically. In practice, the threshold τ is derived from a user-defined target rejection rate (trr) which is the rejection rate of the target class training data.

The one-class classifier is built as follows: at first a class with artificial examples is generated, whereas the feature values obey a Gaussian distribution with $\mu = 0$ and $\sigma^2 = 1$. We employ the Gaussian distribution in favor of a more complex reference distribution to underline the robustness of the approach. The proportion of the generated examples is 0.5 compared with the target class. As class probability estimators, we apply bagged random forest classifiers with 1 000 decision trees and ten bagging iterations. A random forest is a collection of decision trees where voting over all trees is run in order to obtain a classification decision [116, 40]. The decision trees of a forest differ with respect to their features. Each tree is build with a subset of $log_2(|features|) + 1$ randomly chosen features; no tree minimization strategy is followed at training time. The learning algorithm stops if either all leaves contain only examples of one class or if no further splitting is possible. Each decision tree perfectly classifies the training sample but because of its low bias the obtained generalization capability is poor [335, 206]. The combination of several classifiers in a voting scheme reduces the variance and introduces a stronger bias. While the bias of a random forest results from several feature sets, the bias of the bagging approach results from the employment of several training samples, and it is considered to be even stronger [39].

5.3.2 Evaluation

We report on experiments to assess the effectiveness of our modeling and classification approach for detecting the two quality flaws shown in Table 5.2, following Anderka, Stein, and Lipka [9]. The evaluation treats the following issues:

- Since a bias may not be ruled out when collecting outlier examples for a classifier's test sample, we investigate the consequences of the two extreme (overly optimistic, overly pessimistic) settings.
- Since Wikipedia editors have different expectations regarding the classification effectiveness given different flaws, we analyze the optimal operating point for each flaw-specific classifier within the controlled setting of a balanced class distribution.
- Since the true flaw-specific class imbalances in Wikipedia can only be hypothesized, we illustrate the effectiveness of the classifiers in different settings, this way enabling users (Wikipedia editors) to assume an optimistic or pessimistic position.

Outlier selection We propose two strategies for outlier selection to compile a two-class test sample. These outlier examples are not used for the training process of the one-class classifier.

Optimistic Setting Use of featured articles as outliers. This approach is based on the hypothesis that featured articles do not contain a quality flaw at all. Under this setting one introduces some bias since featured articles cannot be considered a representative sample of Wikipedia articles.

Pessimistic Setting Use of a random sample from $D \setminus D_i^-$ as outliers for each f_i, where D_i^- is the population of articles that are not tagged with f_i. This approach may introduce considerable noise since $D \setminus D_i^-$ is expected to contain untagged articles that suffer from f_i.

The above settings address two extremes: classification under laboratory conditions (overly optimistic) versus classification in the wild (overly pessimistic). The experiment design is owing to the facts that "no-flaw features" cannot be stated and that the number of false positives, as well as the number of false negatives in D^- of tagged articles are unknown.

Experiment: Effectiveness of flaw classification In the optimistic setting, outliers are randomly sampled from the 3 128 featured articles. In the pessimistic setting, outliers are randomly sampled from untagged Wikipedia articles, particularly, for each flaw $f_i \in F$ from $D \setminus D_i^-$. We evaluate our approach under both settings by applying the following procedure: for each flaw $f_i \in F$, the corresponding one-class classifier h_i is evaluated in a 10-fold cross-validation

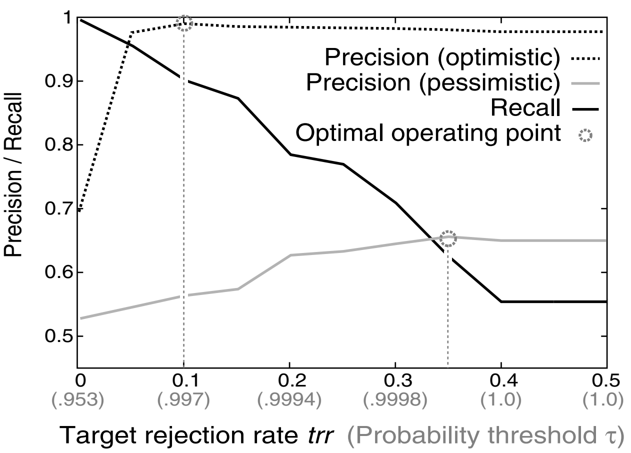


Figure 5.7: *Precision and recall over target rejection rate for the flaw "Unreferenced". The figure illustrates the difference in terms of precision under the optimistic setting, using featured articles as outliers, and the pessimistic setting, using random articles as outliers. The recall is the same under both settings. The optimal operating points correspond to the target rejection rates that maximizes classifier precision. From Anderka, Stein, and Lipka [9].*

setup with 1 000 flawed articles randomly sampled from D_i^-; within each cross-validation run, the classifier h_i is trained on 900 articles from D_i^-, whereas testing is performed with 100 articles from D_i^- plus 100 outliers. Note that h_i is trained exclusively with the examples of the respective target class (articles in D_i^-). The training of h_i is neither affected by the class distribution nor by the outlier-selection strategy that is used in the respective setting.

The precision of the one-class classifier is controlled by the target rejection rate. We empirically determine the optimal operating point for each flaw under the optimistic and pessimistic settings. The optimal operating point corresponds to the target rejection rate of the maximum precision classifier; alternative thresholding methods are described by Shanahan and Roma [271]. Figure 5.7 illustrates the operating point analyses for the flaw "Unreferenced"; with increasing target rejection rate, the recall value decreases while the precision values increase. The recall is the same in both settings, since it solely depends on the target class training sample. For the flaw "Unreferenced", the optimal operating points under the optimistic and pessimistic setting are at the target rejection rates of 0.10 and 0.35 (with precision 0.99 and 0.63).

Table 5.3 shows the effectiveness of flaw prediction. The values correspond to the effectivenesses at the respective optimal operating points. The employed measures are precision, recall, and area under ROC curve [81], which is important to assess the trade-off between specificity and sensitivity of a classifier. An AUC value of 0.5 means that all specificity-sensitivity-combinations are equivalent, which in turn means that the classifier is random guessing.

Table 5.3: *Individual effectivenesses of two flaw classification solutions at the optimal operating point, using featured articles as outliers (optimistic setting) and using random articles as outliers (pessimistic setting). The class distribution is balanced under both settings. The flaw ratio 1:n (flawed articles : flawless articles) corresponds to the estimated actual frequency of a flaw. From Anderka, Stein, and Lipka [9].*

Flaw name	**Optimistic setting**			**Pessimistic setting**			**Flaw**
	Prec	*Rec*	AUC	*Prec*	*Rec*	AUC	**ratio**
Unreferenced	0.99	0.90	0.95	0.63	0.63	0.63	1:3
Advert	0.86	0.91	0.88	0.65	0.58	0.63	1:136

5.3.3 Summary

We examine the classification of information quality flaws in Wikipedia as an example for one-class classification problems in information retrieval. This problem is typical because it is caused by a dynamic and noisy complementary class. To improve the reported effectivenesses, the text representations could be enhanced by new features, and the one-class classification approach could be specifically parametrized.

5.4 Improving one-class classification via ensembles

Text classification tasks that are one-class problems at heart have often a target class distribution with multiple modes. For this case, we propose one-class ensembles that learn each mode separately. In contrast to standard bagging and boosting techniques [272], multiple classifiers are trained on a structured partitioning of the training sample. Each partition, which results from a clustering of the given training sample, represents a particular mode instead of the entire target distribution. For each mode, a single one-class classifier is trained. Our one-class ensemble is a meta-classification method, which allows the integration of arbitrary one-class classification technology.

5.4.1 One-class classification for text classification tasks

Again, in a one-class problem one is given information of the target class only. The task is to define a boundary that encloses as many target examples as possible while minimizing the chance of accepting examples from outside the target class, so-called outliers [303]. An example for a one-class problem is authorship verification [153], where we are given writing examples for a single author T, and we are asked whether a text of unknown authorship has been written by T as well, cf. Sections 4.5 and 4.6. Despite the fact that a sheer endless number of outliers are at our disposal, we are not able to define a closed

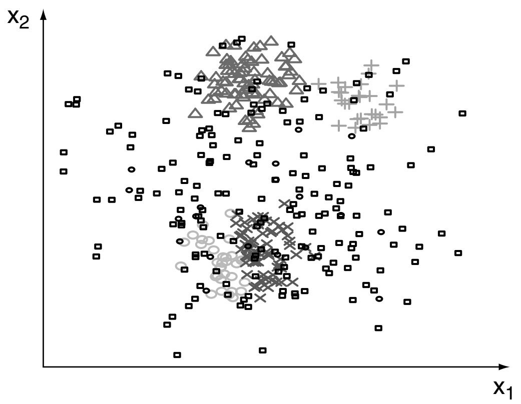


Figure 5.8: *Artificially generated dataset with a diverse target class, constructed as a mixture of Gaussians with one mode for the outlier class and three for the target class. The black squares are outliers, all other documents belong to the target class. The shape and gray scale of the documents indicate a k-means clustering with* $k = 4$.

outlier class with texts from other authors. Specialized one-class classifiers can cope with this setting; however, there is no universal solution for one-class problems, and additional constraints may render the classification task even more challenging. Two of which are common in information retrieval: (1) a highly diverse target class that has a complex, multimodal distribution, and (2) the presence of noise in the application situation of the one-class classifier. We proposed a cluster-based one-class ensemble that is able to effectively alleviate both problems.

Combining binary or multiclass classifiers within an ensemble is proven to increase accuracy in many applications, compared to the best individual classifier in the ensemble [148]. With regard to one-class problems, only few ensemble strategies have been proposed, which can be distinguished into two categories. First, approaches that divide the feature space and train individual one-class classifiers on the different feature subsets [304, 229]. Second, approaches that divide the target class sample and train individual one-class classifiers on the different object subsets [324, 272]. Our approach belongs to the second category and is related to the work of Wang et al. [324], who employ an agglomerative hierarchical clustering strategy in order to partition the target class sample. We employ a suitable clustering technology, apply our approach to real-world information retrieval tasks, and empirically demonstrate its advantage compared with a common one-class SVM.

5.4.2 Cluster-based one-class ensemble

Let $\mathbf{S}_T = \{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$ denote the sample of n feature vectors when given n target examples. The construction of the ensemble classifier happens within four steps:

Step (1): Clustering $\mathbf{S}_T$ is clustered using the k-means algorithm by Hartigan and Wong [107]. The algorithm aims to find an exclusive clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$, $C_j \subseteq \mathbf{S}_T$, $j \leq k \leq n$, such that the variance in each cluster C_j is minimized. The only free parameter in this step is the cluster number k. Figure 5.8 shows an example clustering.

Step (2): One-class classification For each cluster $C_j \in \mathcal{C}$ a one-class SVM $h_j : \mathbf{d} \rightarrow \{+1, -1\}$ is learned. Here, a kernel function ϕ maps the documents into a higher-dimensional feature space. The goal is to find a maximum-margin hyperplane in the kernel space that separates $(1 - v) \cdot n$ target examples from the origin. This is formulated as optimization problem [263]:

$$\begin{array}{rl} & \min_{\mathbf{w},\xi,\rho} \frac{1}{2}||\mathbf{w}||^2 + \frac{1}{v \cdot n}\sum_{i=1}^{n} \xi_i - \rho \\ \text{s.t.} & (\mathbf{w}^T\phi(\mathbf{d}_i)) \geq \rho - \xi_i \text{ with } \xi_i \geq 0,\ i = 1,..,n, \end{array}$$

where $\mathbf{w}$ denotes the normal vector of the hyperplane, ρ the margin, and ξ_i the slack variables. The value $v \in (0, 1]$ is specific to a one-class SVM as it defines the fraction of target examples outside the target class and controls the number of support vectors. The decision function h_j is of the following form:

$$h_j(\mathbf{d}) = sign((\mathbf{w}^T\phi(\mathbf{d})) - \rho)$$

Step (3): Aggregation The ensemble classifier e_k combines the decisions of the k single classifiers for a vector $\mathbf{d}$ as follows:

$$e_k(\mathbf{d}) = \begin{cases} 1 & \text{if } \exists\, h_j(\mathbf{d}) > 0, j = 1, \ldots, k \\ -1 & \text{otherwise.} \end{cases}$$

Step (4): Model selection The clustering parameter k runs from 1 to l, and altogether $l(l+1)/2$ one-class SVMs are constructed. We choose the ensemble e_k that has the lowest classification error on a holdout validation sample.

5.4.3 Analysis and results

In the evaluation, we use a random subset $\mathbf{S}_{train} \subset \mathbf{S}_T$ for the training phase; the test sample $\mathbf{S}_{test}$ comprises a balanced number of target examples from $\mathbf{S}_T \setminus \mathbf{S}_{train}$ and outliers. Each experiment is repeated 15 times. The effectiveness of the classifier is reported as averaged F-measure and for varying values of the number of clusters k. In all experiments a one-class SVM with a non-linear RBF kernel is utilized. The parameters of the classifier h_j are optimized on the respective training sample $C_j \subseteq \mathbf{S}_{train}$; clusters with less than five elements are discarded.

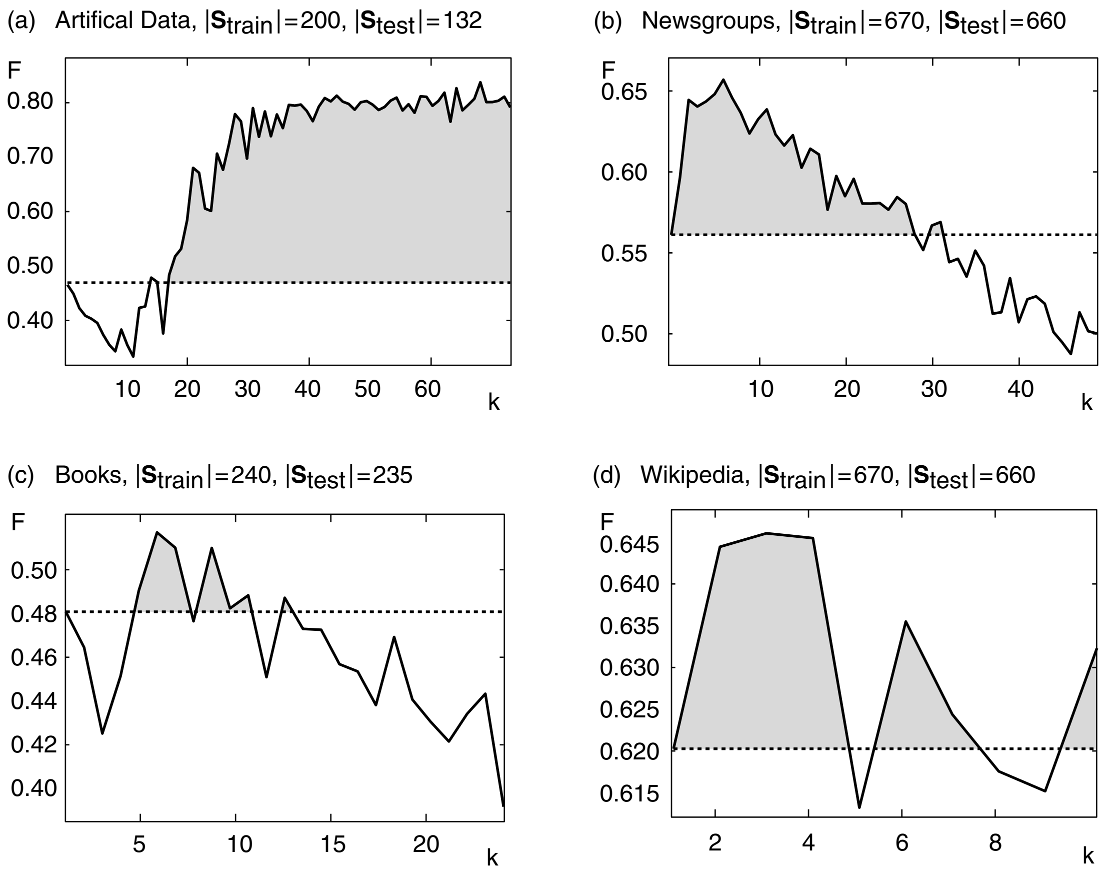


Figure 5.9: *Effectiveness of the cluster-based one-class ensemble approach in terms of the F-measure over the number of clusters k on four datasets. The dotted line shows a one-class SVM, trained on all target examples.*

Figure 5.9 illustrates the results on four different datasets: artificially created documents with three clusters (Figure 5.8), documents from the 20 Newsgroups dataset with computer category in the role of the target class, books from different authors for which the authorship is to be verified [153], and Wikipedia articles tagged with certain quality flaws that are to be detected [9]. All documents are represented under a vector space model with a *tf·idf* weighting, except for Wikipedia articles where quality-specific features are employed [9].

Table 5.4 summarizes the achieved significant improvements over the baseline, which is a one-class SVM that has been trained on all target documents, or equivalently, where $|\mathcal{C}| = 1$. On all four datasets, our approach outperforms the baseline. Considering the real-world tasks, the filtering of news articles in the 20 Newsgroups dataset befits most from the one-class classification ensemble.

5.4.4 Summary

Both, the employed *k*-means algorithm and density estimation or our proposed one-class ensemble can be computed in parallel. Therefore, it can be distributed for example on a Hadoop cluster and is applicable on huge datasets. When using

Table 5.4: *Percentage of improvement over the baseline for each dataset and for the optimum number of clusters k.*

Artificial	Newsgroups	Books	Wikipedia
+70% ($k = 70$)	+18% ($k = 8$)	+7% ($k = 6$)	+4% ($k = 3$)

the one-class ensemble in practice, the best working k has to be validated on a holdout test sample since a higher value of k is not always the best solution. The effectiveness drops for a large number of clusters because the average number of training examples in the clusters shrink and get too small for learning at some point.

5.5 Bibliography

Lack of representativeness The lack of representative training examples is a symptom of one-class and imbalanced classification problems. Reasons for unrepresentative samples are based on the nature of the task, where one or more classes of the classification task have a dynamic distribution or do not exist during the sampling, or based on scarce labeling resources and tools.

We have discussed one-class classification problems in Section 4.6 and 5.3. Recall, when learning to classify flaws in Wikipedia, the flaws are well-defined (closed class) whereas proper articles are arbitrary (open class). A further discussion on one-class classification can be found in Mazhelis [197], Khan and Madden [144]. In addition, most one-class classification tasks can also be tackled by learning from positive and unlabeled (PU) examples if a large fraction of unlabeled examples is available, possibly the entire population [181]. Effective PU learners are based on a generic two-step algorithm:

Step (1) Identify reliable negative examples.

Step (2) Learn from (labeled) positive and reliable negative examples.

The classifier in Step (2) is applied in Step (1) and proceeds iteratively. However, for true one-class classification problems, the positive class is stable (training and test samples are identically distributed), but this is not guaranteed for the complementary (negetive) class. Therefore, PU learning or negative training examples can be harmful to text classification [172].

In contrast, for imbalanced text classification problems, we often face the opposite situation [102]. For example, when few examples are positive, but negative examples are abundant in the training sample. Active learning, as in Section 5.2, can be employed as a mean against imbalance. Under the condition that the local ratio of positive and negative examples close to the actual decision boundary is balanced, an uncertainty sampling strategy would build a better balanced training sample, which is illustrated in [78, 79]. Building proper training samples is also studied in the field instance selection by Liu and Motoda [183]. It should be noted that an SVM implements implicitly an example-selection strategy. Only the examples close to the decision boundary, the support vectors, influence the hypothesis. Therefore, support vectors have a similar effect with respect to balancing compared with uncertainty sampling. In some cases, an oversampling of the minority class [55] or an undersampling of the majority class [188] is appropriate. Finally, the most common research field related to class imbalance is cost-sensitive learning, with the goal to consider examples of the minority class more important by a larger impact on the error function [175, 187].

Presence of noise The quality of the training examples has a similar impact on the effectiveness of a text classification solution as the solution's inductive bias. Related work studies the measurement of sample quality [325], the identification of suspicious examples [159, 356, 45], and the correction or elimination of noisy examples [306, 355].

Measuring the data quality and improving it via correction methods is only applicable to some degree. Therefore, as an alternative to tackle this problem, the learning algorithm has to be able to learn from noisy training examples [113, 354, 357, 25]. The inductive bias of a learning algorithm is directly connected to its noise sensitivity because a high inductive bias can avoid overfitting noisy examples. Atla et al. [15] studied classifiers under the influence of noise: naïve Bayes is more robust to noise than decision trees, SVMs, and logistic regression when the noise level is above 40%; otherwise, decision trees are superior. This outcome is supported by Gamberger et al. [93] who show that the elimination (not the correction) of noisy examples does not affect the learning process of decision trees that apply a pruning strategy. A classifier with a low inductive bias, for example a 1-nearest neighbor classifier, is misled by erroneous examples [336]; a larger neighborhood antagonizes this.

Chapter 6

Model selection for text classification in the wild

The reliable evaluation of classification solutions is a basic necessity, be it for model selection purposes or the assessment of effectivenesses with respect to a given task. Today's information filtering and retrieval tasks [327, 76, 140, 238] deal with classifying texts in volatile environments and render their evaluation more complicated. Large parts of statistical evaluation and machine-learning research rely on the assumption that the provided and the future examples are independent and identically distributed (i.i.d.) with regard to the same underlying probability distribution. It is known that this is not often the case in real-world scenarios, for example, if texts from a time-varying stream are to be classified.

User-generated content on the Web such as news articles, blog posts, and tweets exhibit large variations of the underlying distribution characteristics; especially Twitter exemplifies the volatile nature of "trendy" topics, as illustrated by Liu et al. [186] with a new interactive visualization technique.

Forman [88] subdivides the phenomena of distribution changes over time, also known as concept drift, into three types:

- class distribution shift: the sample of a class remains i.i.d., but the ratio between the classes varies
- subclass distribution shift: the sample of a subclass remains i.i.d., but the sample of the class and the classes overall does not
- fickle concept drift: the ground truth of the class labels changes

The second type, a subclass distribution shift, models the dynamics in online media best and forms the basis of our contribution; it is also known as covariate shift [24] if the shift occurs across subclass boundaries. It is important to note that a subclass distribution shift moreover occurs also if

- the distribution of the population is unknown and therefore different to the training sample, or if
- the training sample is noisy.

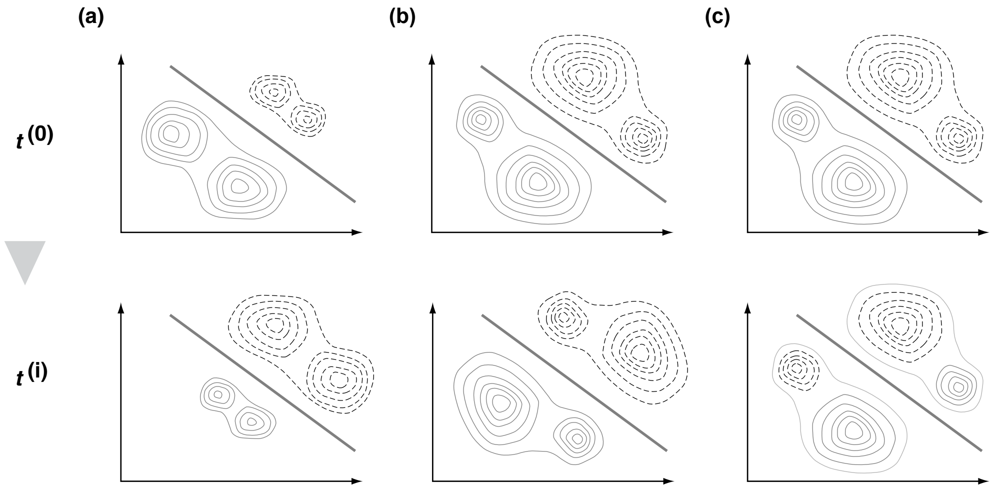


Figure 6.1: *Each of the six plots shows a two-class classification situation (gray, solid versus black, dotted). Three types of distribution changes are illustrated: (a) class distribution shift, (b) subclass distribution shift, and (c) fickle concept drift. The distribution at training time is shown in the upper row ($t^{(0)}$) and the distribution after the change is shown in the lower row ($t^{(i)}$). It should be noted that these types of distribution changes are illustrated in isolation but may occur in a combined fashion as well.*

All of these senarios are typical for text classification tasks in the wild.

This chapter addresses the problem of *evaluating* classification solutions if subclass distribution shifts are likely to occur and if one has no knowledge about how a shift will evolve. Research in semi-supervised learning, domain adaptation [236], and sampling bias correction are related to our problem, but the most approaches assume that knowledge of the target distribution (the distribution for which a classification solution is applied) is given. Other research, which disregards the target distribution, focuses on machine learning within concept drifts and visualization [88, 24].

We propose an evaluation framework that accounts for the nature of text classification in the wild by estimating the expected effectiveness of a classification solution under subclass distribution shifts. For example, in online media such as news streams, articles from the past remain retrievable while a new article substream emerges whenever a new topic becomes interesting. The evolution of the article distribution is hence not arbitrary, and one can expect density changes within local regions that lead to a subclass distribution shift.[1] Currently there is no means for a reliable evaluation nor for a model selection in this scenario, even if the changes of the target distribution are of a homogeneous nature.

Our evaluation framework adapts common statistical evaluation measures to estimate the expected effectiveness and to select the best classification solution in

[1] Also the growth rate of high density areas over the time is not random: Yang and Leskovec [342] studied the dynamics of attention of content within online media and identified six dominant temporal patterns.

terms of the lowest worst case effectiveness. Given a classification solution m we partition the test sample and evaluate m on each partition. The partitioning is constructed by a clustering algorithm that identifies regions of similar examples. The examples in a cluster are likely to behave similarly, a fact which is known as cluster hypothesis in information retrieval: "closely associated documents tend to be relevant to the same requests" [316]. We create new test samples by varying the ratio between the clusters, and, based on these variations, we estimate the expected effectiveness. In addition, we consider the effectiveness of each cluster in isolation and study the effectiveness distribution over the clustering of the test sample. By assuming a constant effectiveness per cluster we derive a second statistics that approximates our idea of expected effectiveness under subclass distribution shifts. Our contributions comprise:

- two statistics to assess the expected effectiveness of a classification solution under subclass distribution shifts,
- a probabilistic notion of the expected effectiveness for model selection,
- an empirical validation of the assertions of our model selection approach for different corpora, and
- an example of the applicability of our expected effectiveness framework for text classification solutions.

6.1 Expected effectiveness under subclass distribution shifts

Depending on the domain and the characteristics of the data, the selection of an appropriate effectiveness measure is a crucial step, particularly in the context of model selection, i.e., preferring one classification solution to another based on their effectiveness. If the classes are imbalanced [322] and the positive class is very small, the accuracy of m, which is the probability of correct classifications, is apparently inappropriate for selecting the better solution. An alternative is to measure the precision, which is the probability of correct positive predictions. In cases where the class balance is unknown, precision becomes difficult to interpret since the number of false positives varies. Recall, however, which is the probability that positive examples are predicted correctly, still provides a straightforward interpretation.

It is assumed in this chapter that the documents are emitted by stochastic processes. It is also assumed that each process is stationary and emits the documents of a subclass independently and identically distributed. Both assumptions qualify for many real-world classification problems. A subclass distribution shift occurs if the emission rates of the processes differ in the course of time. Note that under a subclass distribution shift, all measures that rely on the confusion matrix fail.

We introduce $\mathrm{E}_t\,[e]$, the "expected effectiveness" of a classification solution m under subclass distribution shifts, as the weighted average of the effectiveness that m achieves under all possible subclass distribution shifts. In situations where the development of the underlying distribution cannot be predicted, the expected effectiveness provides a sensible means for model selection. The exact computation of the expected effectiveness is not possible since the underlying emission processes cannot be controlled to produce all possible distribution shifts.

6.2 An expected effectiveness estimate

We estimate the expected effectiveness $\mathrm{E}_t\,[e]$ of m under subclass distribution shifts by identifying subclasses of the underlying stochastic processes and by modeling different distribution shifts via resampling. We associate subclasses with the clusters of a clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$, $C_i \subseteq \mathbf{S}$, $i = 1, \ldots, k$, where $\mathcal{C}$ is an exclusive and complete partitioning of the feature vectors in the sample $\mathbf{S}$. The difference between a clustering on the one hand and a categorization by humans on the other is that the latter is based on the interpretation of real-world objects, while a clustering analyzes densities (DBSCAN [80], MajorClust), variances (k-means, Ward's method [328]), or distributions (expectation-maximization clustering) of feature vectors. If the documents within a cluster are considered to be realizations of a single stochastic process, it is likely that this process emits documents in a high-similarity region of the population.

Distribution shifts are modeled by resampling the documents within the clusters. An increase or decrease of the documents in a high-similarity region (as specified by a cluster) implies that the probability density function of the *global* probability distribution of documents and class labels will change. If, for example, the density values of the global probability density function increase inside a specific region, the density values outside will decrease due to normalization. In our considerations, we constrain the modeling of distribution shifts by preserving the local (cluster-specific) characteristics of the distribution. Stated another way, the probability distribution of a cluster C, $P_C(Y|X)$, remains i.i.d., and the distribution of clusters sizes varies.

Given a clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$, let $\mathcal{S}$ be a set of samples where each $\mathbf{S} \in \mathcal{S}$ is compiled by a unique weighting over $\mathcal{C}$:

$$\mathbf{S} \;=\; \mathit{sample}(C_1) \cup \cdots \cup \mathit{sample}(C_k).$$

The estimate $\widetilde{e}$ for the expected effectiveness $\mathrm{E}_t\,[e]$ of m under subclass distribution shifts is defined as the sample mean over $\mathcal{S}$:

$$\widetilde{e} \;=\; 1/|\mathcal{S}| \sum_{\mathbf{S} \in \mathcal{S}} e(m, \mathbf{S}). \tag{6.1}$$

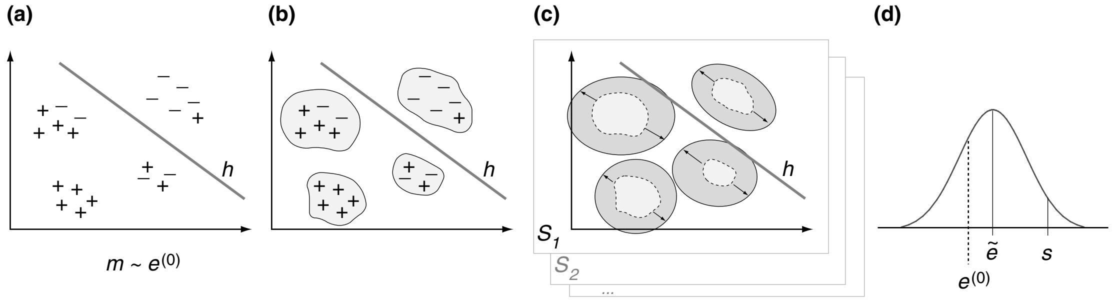


Figure 6.2: *The estimation of the expected effectiveness of a classification solution m (with classifier h) happens in the following steps: (a) input for the estimation, which consists of m and a given test sample of labeled examples, (b) cluster analysis of the test sample in order to identify the modes of the distribution, (c) variation of the cluster sizes and application of m, (d) sample statistics (mean $\widetilde{e}$ and standard deviation s) of the achieved effectivenesses.*

The sample variance s^2 of e can hence be written as:

$$s^2 \;=\; 1/|\mathcal{S}| \sum_{\mathbf{S}\in\mathcal{S}} (e(m,\mathbf{S}) - \widetilde{e})^2. \tag{6.2}$$

The symbol $\widetilde{e}$ is used instead of $\bar{e}$ to emphasize that the mean is computed from a specifically constructed set $\mathcal{S}$.

Figure 6.2 illustrates the estimation procedure:

(a) *Input.* Given is a classification solution m and a test sample. The text representation is defined by the feature engineering α; h is a classifier built by a learning algorithm on a separate training sample, which is not illustrated in the figure.

(b) *Clustering of the test sample.* The given test sample is clustered into h clusters (here, four clusters). Each cluster can comprise documents with different class labels.

(c) *Sample set construction and effectiveness estimation.* A set $\mathcal{S}$ of samples is constructed by randomly resizing the clusters via resampling. For each sample $\mathbf{S} \in \mathcal{S}$ the effectiveness $e(m,\mathbf{S})$ of m is estimated.

(d) *Output.* Finally, the resulting distribution of e over the set $\mathcal{S}$ of samples is computed. $\widetilde{e}$ is the expected effectiveness under a subclass distribution shift.

6.3 An expected effectiveness heuristic

The estimate $\widetilde{e}$ of the expected effectiveness is based on a sufficiently large set $\mathcal{S}$ of samples. We now devise a second statistics $\widehat{e}$ for $\mathrm{E}_t\,[e]$, which considers only the characteristics of the clustering $\mathcal{C}$ of the test set $\mathbf{S}$.

In this regard we evaluate for each cluster $C \in \mathcal{C}$, $|\mathcal{C}| = k$, its cluster-specific effectiveness e_C of m. Since the effectiveness is likely to be the same on similar documents, it can be assumed that m's effectiveness for a cluster C remains stable under a subclass distribution shift. This assumption relates to the clustering assumption in the field of semi-supervised learning: "If points are in the same cluster, they are likely to be of the same class" [54]. Recall the clustering hypothesis: closely associated documents tend to be relevant to the same requests. In terms of classification this can be interpreted as closely associated documents tend to have the same class label.

The (overall) effectiveness e of m given $\mathbf{S}$ is the weighted sum of the cluster-specific effectiveness values:

$$e = w_1 e_{C_1} + \cdots + w_k e_{C_k},$$

where w_i is the weight given by the relative size of the cluster $|C_i|/|\mathbf{S}|$. Using vector notation, with a positive real-valued weight vector $\mathbf{w}, |\mathbf{w}| = k$, and effectiveness vector $\mathbf{e} = (e_{C_1}, \ldots, e_{C_k})^T$, the effectiveness is $e = \mathbf{w}^T\mathbf{e}$, where $\mathbf{w}^T$ denotes the transpose of $\mathbf{w}$.

We now assume that the effectiveness for a cluster shows no variation at different points in time:

$$\mathbf{e} \equiv \mathbf{e}^{(0)} = \mathbf{e}^{(1)} = \ldots \tag{6.3}$$

Since this assumption depends on the degree to which the clustering assumption is fulfilled, we call the statistic $\widehat{e}$ for the expected effectiveness $\mathrm{E}_t\left[e\right]$, introduced below, a heuristic.

Note that Assumption (6.3) does not imply a constant (overall) effectiveness e. Under Assumption (6.3), the effectiveness e varies only with the change of the weights $\mathbf{w}$. We model the weight vector $\mathbf{w}$ as a k-dimensional random variable $W = (W_1, \ldots, W_k)$. Without knowledge about future subclass distributions, our risk-minimization strategy is to consider all possible vectors $\mathbf{w}$ as equally likely, whereas the $ell1$-norm of $\mathbf{w}$ is always one: $\sum_{i=1}^{k} w_i = 1$. As a consequence, the vectors $\mathbf{w}$ lie on a simplex and W is Dirichlet distributed, $W \sim Dir(\boldsymbol{\alpha})$, with concentration hyperparameter $\boldsymbol{\alpha} = (1, \ldots, 1)^T$, $|\boldsymbol{\alpha}| = k$. The resulting mean and variance are:

$$\begin{aligned} \mathrm{E}\left[W_i\right] &= \frac{\alpha_i}{\alpha_0} = 1/k, \quad \text{where } \alpha_0 = \sum_{j=1}^{k} \alpha_j, \\ \mathrm{Var}\left[W_i\right] &= \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)} = \frac{k-1}{k^2(k+1)}. \end{aligned}$$

Let e' denote the random variable $W_1 e_{C_1} + \cdots + W_k e_{C_k}$. Due to the central limit theorem, e' is normally distributed. Therefore, the heuristic $\widehat{e}$ of the expected

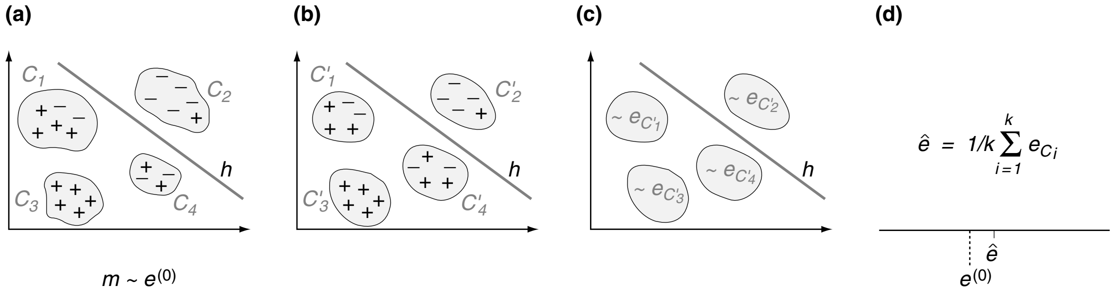


Figure 6.3: *Heuristic estimation of the expected effectiveness of m: (a) application of cluster analysis to the test sample in order to identify the modes of the distribution (this corresponds to Step (b) in Figure 6.2), (b) adaptive sampling of the clusters in order to equalize different cluster sizes that may be given in the sample, (c) application of m for each cluster in isolation, (d) mean $\widehat{e}$ of the cluster-specific effectiveness values.*

effectiveness $\mathrm{E}_t\,[e]$ of m under subclass distribution shifts is the mean of e':

$$\begin{aligned}\widehat{e} &= \mathrm{E}\,[W_1]\,e_{C_1} + \cdots + \mathrm{E}\,[W_k]\,e_{C_k} \\ &= 1/k\; e_{C_1} + \cdots + 1/k\; e_{C_k} \;=\; 1/k \sum_{i=1}^{k} e_{C_i}. \end{aligned} \qquad (6.4)$$

The variance of the random variable e' is:

$$\mathrm{Var}\left[e'\right] \;=\; \frac{k-1}{k^2(k+1)}.$$

$\mathrm{Var}\,[e']$ is independent of the cluster-specific effectiveness values and hence constant for all classification solutions. Hence, this variance cannot be exploited for model selection purposes. We conclude that the difference between $\mathrm{Var}\,[e']$ and the sample variance s^2 of e (Equation 6.2) reflects to what extent Assumption (6.3) is violated, say, to what extent the effectiveness within the clusters is not constant.

Assumption (6.3) is strict, but not unrealistic, and we can show in the experimentation section that our heuristic has in practice only a small approximation error (about 3%) for estimating the expected effectiveness under a subclass distribution shift.

Figure 6.3 illustrates the computation of the heuristic:

(a) *Input.* A classification solution m along with a clustering $\mathcal{C}$ of the test sample with k clusters.

(b) *Adaptive sampling.* The k clusters are scaled to the same size, $|C'_1| = \cdots = |C'_k|$, to get better effectiveness estimates for each cluster.

(c) *Effectiveness estimation.* For each cluster the effectiveness $e_{C'}$ of m on C' is estimated.

(d) *Output.* $\widehat{e}$, the mean of the $e_{C'}$, which represents a heuristic estimate of the expected effectiveness under distribution shifts in the clustering (Equation 6.4).

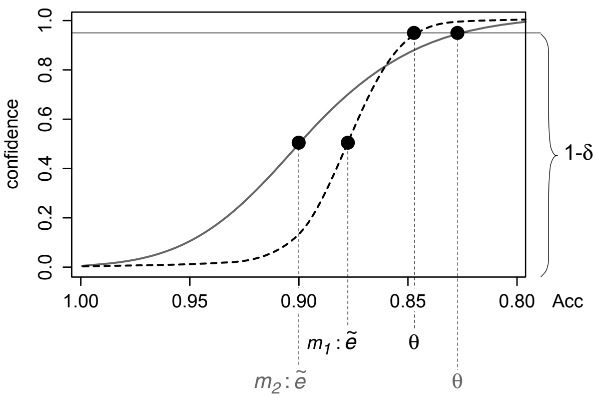


Figure 6.4: *Comparison of classification solution* m_1 *(*$\widetilde{e} = 0.88, s = 0.02$*, dotted line) to classification solution* m_2 *(*$\widetilde{e} = 0.90, s = 0.04$*, solid line) with respect to their accuracies. At confidence level* 0.5*, solution* m_2 *has a higher expected accuracy than* m_1*. At confidence level* $\delta = 0.05$*, however, the lower bound (worst case)* θ *under* m_1 *is better than* θ *under* m_2*.*

6.4 Model selection via expected effectiveness

Given a clustering $\mathcal{C}$ of a test set $\mathbf{S}$, model selection means to choose a classification solution m from a set of solution candidates M. If the expected effectiveness $\mathrm{E}_t\,[e]$ is approximated under Assumption (6.3) as $\widehat{e}$, the model selection problem can be tackled by choosing the model with the highest $\widehat{e}$. If the expected effectiveness is approximated via the estimation procedure (Figure 6.2) as $\widetilde{e}$, additional model selection information in form of a probabilistic lower effectiveness bound θ can be provided.

We present such a lower bound θ to show that the effectiveness of classification solution m is with a probability of $1 - \delta$ larger than θ, if the subclass distribution varies. The effectiveness e is normally distributed, see Section 6.3, with approximated mean $\widetilde{e}$ and variance s^2 (Equations (6.1) and (6.2)). We can estimate the parameters of the normal distribution for each solution in M and infer θ with the inverse of its cumulative distribution function, also known as quantile function:

$$\begin{aligned} F^{-1}(\delta;\, \mu, \sigma^2) &= \mu + \sigma\sqrt{2}\ \mathrm{erf}^{-1}(1 - 2\delta) \\ \theta &= F^{-1}(\delta;\, \widetilde{e}, s^2), \end{aligned} \tag{6.5}$$

where F^{-1} denotes the quantile function of the normal distribution for $1 - \delta$, and erf^{-1} denotes the inverse error function. If δ is chosen to be 0.0228, the value of θ is $\widetilde{e} + 2s$. Figure 6.4 shows an example for the accuracy *Acc*.

While the expected effectiveness estimate is useful for selecting the classification solution with the best expected effectiveness, the probabilistic lower bound is useful for selecting the solution that minimizes the risk of an effectiveness drop in the wild.

6.5 Validation of prior assumptions

The experimentation section validates our theoretical findings and demonstrates the utilization of the notion of expected effectiveness. We empirically show for various corpora, classifier, and measures, that

- the expected effectiveness under a shifting subclass distribution is normally distributed, and therefore,
- the expected effectiveness can be applied within a probabilistic lower bound for model selection, and finally that
- the heuristic expected effectiveness $\widehat{e}$ is a tight estimate.

In order to demonstrate the evaluation of classification solutions under subclass distribution shifts we will focus on standard text classification corpora and standard learning algorithms. As text classification task the topic categorization problem is studied: given a set of topics, assign an unseen document to one of these topics.

6.5.1 Data sources

Reuters (RVC1) [171], the Open Directory Project (ODP)[2], and 20 Newsgroups (20NG)[3] are the most frequently used datasets within the field of topic categorization. Our preprocessing of the corpora restricts them to documents that are uniquely classified and that have a minimum size of 1 kB. We construct binary classification tasks by selecting two categories instead of pursuing a one-vs-all strategy. From the large number of tasks that have been considered in our experiments, we will report results for the task Science vs Sports only since the categories occur across all corpora.

RVC1 The Reuters corpus contains about 810 000 news documents, categorized in 4 top-level categories. We have sampled 4 000 documents from the categories Sports (topic code: GSPO) and Science & Technology (topic code: GSCI).

ODP The Open Directory Project is a human-edited directory of the Web and provides an RDF dump of the links within each category. We have downloaded about 15 000 000 webpages within 14 out of 16 top-level categories and extracted the plain texts. 200 000 documents are sampled from the categories Science (subcategories: Environment, Math, Biology, Physics, Technology) and Sports (subcategories: Hockey, Basketball, Tennis, Baseball, Football). About 92 % of the leaf categories are represented and the average maximal depth is 6.45.

[2] http://www.dmoz.org/

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

20NG The 20 Newsgroups dataset contains about 20 000 newsgroup documents in 20 categories. We have removed duplicates and parsed the plain text by ignoring all headers. 6 000 documents are selected from the categories Science (subcategories: Electronics, Space, Cryptography, Medicine) and Sports (subcategories: Baseball, Hockey).

6.5.2 Classification solutions

As mentioned at the outset, classification solutions comprise a text representation α and a classifier h. We vary the range of solutions by employing several learning algorithms, while α remains unchanged.

The text representation is the following: The documents of each considered corpus are represented under a vector space model with word frequencies whereas the dimensions correspond to the stemmed alphabetic words that occur at least 10 times. The Lovins stemmer is employed as stemming technology and the vectors are normalized. To run a vast amount of experiments within a reasonable time, 2 500 words with the highest information gain scores remain after further processing, while the scores are evaluated only on the training examples with discretized word frequencies.

The following learning algorithms are employed in the experiments to compile h:

- a linear support vector machine (SVM)
- a naïve Bayes classifier (NB)
- a decision tree (C4.5)
- a k-nearest neighbor classifier (k-NN)

6.5.3 Clustering algorithm

The comparison of different clustering algorithms is beyond the scope of this paper. The appropriate choice depends on the application domain and the concrete classification task. It cannot be expected that a single algorithm is always the best choice (no free lunch). Within our experiments the k-means algorithm is applied whereas k is selected heuristically. The k-means algorithm generates an non-overlapping (exclusive) clustering, i.e., $\forall_{i,j,i \neq j} C_i \cap C_j = \emptyset$. Formally, the goal of k-means is to find a clustering that minimizes the following objective function:

$$\underset{\mathcal{C}}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{\mathbf{d}_j \in C_i}^{|C_i|} \|\mathbf{d}_j - \boldsymbol{\mu}_i\|^2.$$

We set k to $\sqrt{|\mathbf{S}|/2}$ and use the Lloyd's algorithm for approximation. For the Reuters corpus RCV1 we explicitly show that this setting is able to identify

Table 6.1: *Randomness of the data emission for three categories of the Reuters corpus, quantified by Bartels and Wald-Wolfowitz tests. The p-values are computed for the entire sets (overall) as well as for the clusterings (avg. cluster).*

Science		Sports		Politics	
avg. cluster p-values	overall p-value	avg. cluster p-values	overall p-value	avg. cluster p-values	overall p-value
Bartels Test					
0.169	≈ 0	0.139	≈ 0	0.141	≈ 0
Wald-Wolfowitz Test					
0.274	≈ 0	0.202	≈ 0	0.105	≈ 0

appropriate subclasses. RCV1 provides time stamps as opposed to ODP and 20NG and has been shown to have a shifting subclass distribution by visualizing the change of the top 100 most predictive words over time [88].

For the RCV1 we explicitly show that the setting is able to identify appropriate subclasses: opposed to ODP and 20NG, the RCV1-corpus provides time stamps. Moreover, by visualizing the change of the top 100 most predictive words over time, Forman [88] gave evidence for a subclass distribution shift in RCV1. With statistical randomness tests, we empirically validate the two main properties of subclass distribution shifts:

(1) the overall samples at different time stamps *are not* i.i.d. according to the same distribution, but

(2) the samples of isolated subclasses *are* i.i.d.

The most practical randomness tests operate on binary sequences, which are often constructed by dichotomizing a sequences of continuous values. For a sample $\mathbf{S}$ we consider the sequences of the Euclidean distances $dist(\mathbf{d}^{(i)}, \mathbf{d}^{(i-1)})$ for $i = 2 \ldots |\mathbf{S}|$, where $\mathbf{d}^{(i-1)}$ is the chronological predecessor of $\mathbf{d}^{(i)}$. A sequence of distances that is i.i.d. according to an unknown distribution indicates a process of data generation that can produce i.i.d. samples. As an illustrative example consider a random process that first produces fairly similar news articles on politics and after a while articles on sports. The corresponding (non-random) sequence of distances is a series of small distances followed by a large distance when the generation of sports articles begins. As a consequence, the samples drawn at different points in time are not representative for the same distribution.

In the following, we test the randomness of the process of generating Reuters articles with the Wald-Wolfowitz runs test and the Bartels test on the chronological sequence of distances. We also test the isolated generation of articles within subclasses defined by clusters. Table 6.1 shows the results of this study. The p-values of the respective tests indicate that null hypothesis (the articles are i.i.d.) is rejected when the data is analyzed as a whole but on average accepted

Table 6.2: *Shapiro-Wilk test for testing whether the expected effectiveness is normally distributed. The W and the p-value are averaged over all classifiers in Section 6.5.2 and all measures in Table 3.3.*

Science vs Sports					
ODP		**RCV1**		**20NG**	
W	p-value	W	p-value	W	p-value
0.93	0.29	0.92	0.17	0.93	0.37

when each cluster is analyzed in isolation, i.e., the articles are possibly i.i.d. generated.

6.6 Application and conclusions

To apply model selection as described in Section 6.4 the effectiveness has to be normally distributed under subclass distribution shifts. We revisit the theoretical result that the effectiveness is normally distributed by employing the Shapiro-Wilk test, which has been shown to be one of the most powerful tests of normality [242, 345]. The value W of the test is the ratio between two variance estimators for a random sample $e_1 < e_2 < \cdots < e_n$. The first variance estimator is the expected variance of an assumed normal distribution while the second variance estimator is the bias corrected variance of the given random sample [242]. A W close to 1 indicates a normal distribution. The high p-value of the Shapiro-Wilk test indicates that the null hypothesis, the data is normally distributed, cannot be rejected. For the results reported in Table 6.2, we removed the 5 highest and lowest values in the evaluation since the test is very sensitive to outliers. For all measures and classifiers the estimated effectiveness passed the test under the subclass distribution shift.

Table 6.3 summarizes our proposed notions applied to the selected classification tasks. We restrict the presentation to the most commonly used measures, namely accuracy *Acc*, precision *Prec*, and recall *Rec*. The expected effectiveness is estimated on 1 000 different generated test samples $\mathbf{S}$ based on the initial clustering. The reported results are averaged over 10 different testing and training samples, which are randomly chosen from the respective corpus. The values for the probabilistic lower bound θ result from $\delta = 0.0228$.

Notice that with a few exceptions the heuristic expected effectiveness is tight. The average approximation is about 3 % when comparing $\tilde{e}$ and $\widehat{e}$.

The values in Table 6.3 serve as an illustration of the enhancement of common measures to provide more insights when evaluating classification solutions. Observing the accuracies of the SVM and NB classifiers on the Reuters corpus, the SVM performs slightly better on the initial test set, which is still the case,

Table 6.3: *Summarization of characteristics of classification solutions. For various classifiers, corpora, and measures the table reports: the initial effectiveness $e^{(0)}$, the heuristic expected effectiveness estimate $\widehat{e}$ (Equation 6.4), the expected effectiveness $\tilde{e}$ (Equation 6.1), and the probabilistic effectiveness bound θ (Equation 6.5) based on $\tilde{e}$ and the sample variance s^2 (Equation 6.2).*

Classifier	**Measure**											
	Acc				*Prec*				*Rec*			
	$e^{(0)}$	$\widehat{e}$	$\widetilde{e}$	θ	$e^{(0)}$	$\widehat{e}$	$\widetilde{e}$	θ	$e^{(0)}$	$\widehat{e}$	$\widetilde{e}$	θ
RCV1												
SVM	0.987	0.995	0.988	0.978	0.99	0.97	0.99	0.98	0.98	0.99	0.99	0.97
NB	0.986	0.993	0.987	0.979	0.98	0.98	0.98	0.97	0.99	0.99	0.99	0.98
C4.5	0.921	0.932	0.934	0.911	0.94	0.82	0.95	0.92	0.90	0.92	0.92	0.89
k-NN	0.965	0.978	0.972	0.951	0.98	0.95	0.98	0.97	0.95	0.98	0.96	0.94
ODP												
SVM	0.962	0.951	0.957	0.926	0.95	0.96	0.94	0.90	0.97	0.95	0.98	0.94
NB	0.927	0.941	0.933	0.899	0.93	0.95	0.92	0.88	0.92	0.93	0.94	0.91
C4.5	0.927	0.919	0.924	0.904	0.92	0.92	0.91	0.86	0.93	0.91	0.94	0.91
k-NN	0.878	0.931	0.883	0.836	0.82	0.84	0.82	0.74	0.96	0.98	0.96	0.93
20NG												
SVM	0.890	0.968	0.906	0.860	0.90	0.94	0.90	0.82	0.88	0.94	0.91	0.85
NB	0.880	0.882	0.905	0.850	0.94	0.88	0.94	0.88	0.82	0.95	0.86	0.75
C4.5	0.790	0.776	0.788	0.720	0.75	0.72	0.73	0.61	0.90	0.91	0.90	0.84
k-NN	0.700	0.788	0.704	0.640	0.75	0.87	0.72	0.63	0.61	0.69	0.65	0.56

when we assume subclass distribution shifts and compare the expected effectiveness. Nevertheless, the probabilistic lower bound for the NB classifier is higher. The expected effectiveness does not necessarily has to be worse than the initial effectiveness. The recall of the considered SVM in the 20NG corpus is expected to be higher than initially estimated.

We presented the notion of expected effectiveness and its probabilistic lower bound as a basis for preferring one classification solution over another when the underlying data source undergoes a shift in the distribution of its subclasses. Subclass distribution shifts occur in many real-world classification applications, and quite often one has no knowledge about how such a shift will evolve. Our idea is to prefer the solution that has the best probabilistic lower bound of its effectiveness. This bound is based on the expected effectiveness if all shifts are considered equally likely.

Our estimate of the expected effectiveness relies on a repetitive resampling of the clustered test sample for different margin distributions. Clustering is an appropriate method, as exemplified in the experimentation section for news articles, for the identification of those subclasses that are not subject to distribution shifts. The effectiveness within these subclasses is nearly constant. This observation suggests a heuristic for computing another expected effectiveness estimate, namely, to use the mean effectiveness over the clustering. In an empirical evaluation we applied the outlined considerations to standard text corpora, and we showed that the heuristic for the expected effectiveness has a low approximation error.

A number of clustering algorithms are available, and we regard the selection of the most appropriate one, as well as the setting of its hyperparameters as the task of the engineer. If the clustering algorithm is able to identify subclasses that are likely to grow with a stationary distribution, the expected effectiveness estimate is reliable. In this regard, one should recall that is not the main goal to find semantically meaningful clusterings; in fact, we suggest preferring fine-granular clusterings to coarse clusterings as long as each cluster contains a sufficient number of documents. In practice, however, it is also not critical from a statistical point of view to remove very small clusters from the clustering.

6.7 Bibliography

Concept drift A concept drift is a change of the distribution of examples over time. Research in this field can be divided into concept drift detection and concept drift handling. The objective is to compile a learning algorithm that detects a drift and adapts to it. Concept drifts occur either gradually or abruptly and are empirically observable in labeled and unlabeled samples [174]. This can be done, for example, by monitoring the prediction quality, the distribution, or clustering parameters such as densities, centers, or shapes. Vreeken et al. [321], for example, estimated the differences between two samples by techniques based on compression and covering characteristics; Anderson et al. [10] compared the distances between density estimates. Standard drift detection employs statistical hypothesis testing for the randomness of samples, such as the Wald-Wolfowitz test or more advanced tests [74].

Concept drift handling became an important research topic in recent years; the most common machine-learning methods were adapted to handle it, and theoretical results were extended to capture concept drift phenomena. Advanced window-based approaches are given in [352] and [217]: the former paper proposes a window-based one-class ensemble, whilst the latter proposes a window-based ensemble for learning from positive and unlabeled examples in order to accurately select and classify unlabeled examples for reuse. Huang [127] extended a sampling function for active learning by monitoring concept drifts in unlabeled data. Aggarwal et al. [4] presented a classification method that adapts to changes of the underlying data stream by dynamically selecting an appropriate training sample. Finally, Hulten et al. [128] focussed on novel decision tree learning algorithms, where outdated subtrees are revisited and recreated.

Domain adaptation Within a source domain, where labeled examples are available, a classification solution m is built with the aim to deploy m to a different target domain (e.g., using m for classifying tweets while m has been trained on news articles). For the target domain, unlabeled and sometimes a small amount of labeled examples are available. The main problem is that the margin and the conditional distributions of the source and target domain may differ.

A simple approach for domain adaptation is to ignore these differences and transform the given problem to a semi-supervised learning problem by considering source and target domain as a whole and by applying semi-supervised algorithms as described in [63, 338].

One principle in domain adaptation is to make margin and conditional distributions of the source domain and the target domain more similar by changing the representation. Advanced approaches (e.g. [27]) augment the origin representation by constructed features and it has been shown that the similarity between source and target distributions is increased [21].

Another principle is to make assumptions about the probability distributions. Under the assumption that the conditional distributions are the same but the class distributions of the source and the target domain differ, Chan and Ng [53] recalibrated the learned probabilities by estimating the shift of the priors. The assumption that the conditional distributions are the same but the marginal distributions differ is known as covariate shift [273] and is treated, for example, in [236].

Sample selection bias Having a biased sample of training examples drawn from the target distribution, a classification solution that is build on this data is expected to perform different in the wild. Standard approaches that tackle the selection bias in a sample commonly use a large amount of unlabeled examples from the target distribution. They are based on a weighting of the training examples, a recalibration of the priors, or a resampling to correct the distribution of the training examples. Again, for most of these approaches, the target distribution has to be known in terms of unlabeled examples [60, 125].

Chapter 7

Conclusions

If one looks at the Web with all its applications—either for consumers, business institutions, or professionals—text classification is omnipresent, be it in a mobile app that filters news articles, in a big data sentiment analysis to monitor public opinions, or in a Web service that assists professional forensic analysts in plagiarism cases. Text classification becomes an effective tool if its modeling process reinforces strengths and diminishes weaknesses of text classification solutions.

This thesis centers on the process of modeling text classification tasks, which governs the effectiveness of the resulting solutions. For a successful model in terms of appropriate representations, sampling strategies, learning algorithms, and model selection criteria, it is crucial for an engineer to understand the foundations of these aspects, as well as the domain of the task. This process has been exemplified for several non-standard text classification tasks during this thesis. Furthermore, new algorithms that can lead to improvements of the classification effectiveness have been proposed and analyzed, including text representations beyond the bag-of-words model, which are appropriate for Web genre, information quality, language, and authorship analyses. What is more, methods of facing machine-learning problems in the wild and a risk-minimizing model selection, when the classification solution is applied under subclass distribution shifts in the wild, have been introduced.

The research in this thesis is limited in so far as it does not provide a general framework for arbitrary text classification tasks. This is the nature of things: in spite of the fact that statistical sampling, computational learning theory, and model selection are well understood, no general algorithms exist that are ideal for all tasks in general (no free lunch). This thesis, however, may encourage the reader to thoughtfully approaching text classification tasks and to conducting tailored experiments for discovering solutions.

The future impact of text classification on real-world tasks corresponds to the advancement in finding suitable hypotheses and theories. The chapters "Towards effective text classification in the wild" (Chapter 5) and "Model selection for text classification in the wild" (Chapter 6) have introduced these issues. The focus of future research should be on relating laboratory conditions, especially closed world assumptions, to reality. The goal is to ensure the applicability of text classification solutions and the interpretability of their evaluation.

References

[1] B. T. Adler and L. de Alfaro. A content-driven reputation system for the Wikipedia. In *Proceedings of the 16th international conference on World Wide Web*, 2007.

[2] C. C. Aggarwal and C. Zhai. A survey of text clustering algorithms. In *Mining Text Data*. Springer-Verlag, 2012.

[3] C. C. Aggarwal and C. Zhai. A survey of text classification algorithms. In *Mining Text Data*. Springer-Verlag, 2012.

[4] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. On demand classification of data streams. In *Proceedings of the 10th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, 2004.

[5] M. Amini and P. Gallinari. The use of unlabeled data to improve supervised learning for text summarization. In *Proceedings of the 25th international ACM SIGIR conference on Research and development in information retrieval*, 2002.

[6] M. Anderka, N. Lipka, and B. Stein. Evaluating cross-language explicit semantic analysis and cross querying. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments*. Springer-Verlag, 2010.

[7] M. Anderka, B. Stein, and N. Lipka. Towards automatic quality assurance in Wikipedia. In *Proceedings of the 20th international conference on World Wide Web*, 2011.

[8] M. Anderka, B. Stein, and N. Lipka. Detection of text quality flaws as a one-class classification problem. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011.

[9] M. Anderka, B. Stein, and N. Lipka. Predicting quality flaws in user-generated content: The case of Wikipedia. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 2012.

[10] N. H. Anderson, P. Hall, and D. M. Titterington. Two-sample test statistics for measuring discrepancies between two multivariate probability density functions using kernel-based density estimates. *Journal of Multivariate Analysis*, 50:41–54, 1994.

[11] P. Antunes, C. Costa, and J. Ferreira Dias. Applying genre analysis to EMS design: The example of a small accounting firm. In *Proceedings of the 7th international workshop on Groupware*, 2001.

[12] N. Archak, A. Ghose, and P. Ipeirotis. Show me the money! Deriving the pricing power of product features by mining consumer reviews. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2007.

[13] S. Argamon, M. Šarić, and S. S. Stein. Style mining of electronic messages for multiple authorship discrimination: first results. In *Proceedings of the 9th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, 2003.

[14] S. Argamon, S. Dhawle, M. Koppel, and J. W. Pennebaker. Lexical predictors of personality type. In *Proceedings of classification society of North America*, 2005.

[15] A. Atla, R. Tada, V. Sheng, and N. Singireddy. Sensitivity of different machine learning algorithms to noise. *Journal of Computing Sciences in Colleges*, 26(5):96–103, 2011.

[16] M. Aufenanger, N. Lipka, B. Klopper, and W. Dangelmaier. A knowledge-based Giffler-Thompson heuristic for rescheduling job-shops. In *Proceedings of the IEEE symposium on Computational Intelligence in Scheduling*, 2009.

[17] L. D. Baker and A. K. McCallum. Distributional clustering of words for text classification. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998.

[18] J. Becker and D. Kuropka. Topic-based vector space model. In *Proceedings of the 6th international conference on Business Information Systems*, 2003.

[19] R. Bekkerman, R. E. Yaniv, N. Tishby, and Y. Winter. Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research*, 3:1183–1208, 2003.

[20] N. Bel, C. Koster, and M. Villegas. Cross-lingual text categorization. In *Research and Advanced Technology for Digital Libraries*. Springer-Verlag, 2003.

[21] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems 19*, 2007.

[22] K. Berkling, T. Arai, and E. Barnard. Analysis of phoneme-based features for language identification. In *Proceedings of the IEEE international conference on Acoustics, Speech, and Signal Processing*, 1994.

[23] Y. Bernstein and J. Zobel. A scalable system for identifying co-derivative documents. In *Proceedings of the String Processing and Information Retrieval symposium*, 2004.

[24] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th international conference on Machine learning*, 2007.

[25] B. Biggio, B. Nelson, and P. Laskov. Support vector machines under adversarial label noise. *Journal of Machine Learning Research*, 20:97–112, 2011.

[26] E. Blanzieri and A. Bryl. A survey of learning-based techniques of email spam filtering. *Artificial Intelligence Review*, 29(1):63–92, 2008.

[27] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, 2006.

[28] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the workshop on Computational Learning Theory*, 1998.

[29] J. E. Blumenstock. Size matters: word count as a measure of quality on Wikipedia. In *Proceedings of the 17th international conference on World Wide Web*, 2008.

[30] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the vapnik-chernovenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

[31] E. Boese and A. Howe. Effects of Web document evolution on genre classification. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, 2005.

[32] H. Borko and M. Bernick. Automatic document classification. *Journal of the ACM*, 10:151–161, 1963.

[33] H. Borko and M. Bernick. Automatic document classification. Part II: Additional experiments. *Journal of the ACM*, 11:138–151, 1964.

[34] R. Bose and J. Frew. Lineage retrieval for scientific data processing: a survey. *ACM Computing Surveys*, 37(1):1–28, 2005.

[35] D. Bourigault. LEXTER, A terminology extraction software for knowledge acquisition from texts. In *Proceedings of the 9th workshop on Knowledge acquisition for knowledge based system*, 1995.

[36] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[37] U. Brandes, P. Kenis, J. Lerner, and D. van Raaij. Network analysis of collaboration structure in Wikipedia. In *Proceedings of the 18th international conference on World Wide Web*, 2009.

[38] A. Bratko, B. Filipič, G. V. Cormack, T. R. Lynam, and B. Zupan. Spam filtering using statistical data compression models. *The Journal of Machine Learning Research*, 7:2673–2698, 2006.

[39] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[40] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[41] I. Bretan, J. Dewe, A. Hallberg, N. Wolkert, and J. Karlgren. Web-specific genre visualization. In *Proceedings of the WebNet World conference on the WWW and Internet*, 1998.

[42] S. Brin, J. Davis, and H. Garcia-Molina. Copy detection mechanisms for digital documents. In *Proceedings of the ACM SIGMOD international conference on Management of data*, 1995.

[43] A. Broder. A taxonomy of Web search. *SIGIR Forum*, 36:3–10, 2002.

[44] A. Broder, N. Eiron, M. Fontoura, M. Herscovici, R. Lempel, J. McPherson, R. Qi, and E. Shekita. Indexing shared content in information retrieval systems. In *Proceedings of the 10th international conference on Advances in Database Technology*, 2006.

[45] C. E. Brodley and M. A. Friedl. Identifying mislabeled training data. *Computing Research Repository*, 2011.

[46] K. Bühler. *Sprachtheorie. Die Darstellungsfunktion der Sprache*. Gustav Fischer Verlag, 1934.

[47] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[48] J. Carbonell. *Subjective Understanding: Computer Models of Belief Systems*. PhD thesis, Yale University, 1979.

[49] R. Caruana, A. Munson, and A. Niculescu-Mizil. Getting the most out of ensemble selection. In *Proceedings of the 6th IEEE international conference on Data Mining*, 2006.

[50] P. Carvalho, L. Sarmento, M. J. Silva, and E. de Oliveira. Clues for detecting irony in user-generated contents: oh...!! it's "so easy" ;-). In *Proceedings of the 1st international workshop on Topic-sentiment analysis for mass opinion*, 2009.

[51] W. B. Cavnar and J. M. Trenkle. N-gram-based text categorization. In *Proceedings of the 3rd annual symposium on Document Analysis and Information Retrieval*, 1994.

[52] J. Chall and E. Dale. *Readability Revisited: The new Dale-Chall Readability Formula*. Brookline Books, 1995.

[53] Y. S. Chan and H. T. Ng. Estimating class priors in domain adaptation for word sense disambiguation. In *Proceedings of the international conference on Computational Linguistics and the annual meeting of the Association for Computational Linguistics*, 2006.

[54] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, 2006.

[55] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[56] J. Cheney, L. Chiticariu, and W.-C. Tan. Provenance in databases: Why, how, and where. *Foundations and Trends in Databases*, 1(4):379–474, 2009.

[57] S.-C. Chin, W. N. Street, P. Srinivasan, and D. Eichmann. Detecting Wikipedia vandalism with active learning and statistical language models. In *Proceedings of the 4th workshop on Information credibility*, 2010.

[58] F. Choi. Advances in domain independent linear text segmentation. In *Proceedings of the 1st conference on North American chapter of the Association for Computational Linguistics*, 2000.

[59] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20: 273–297, 1995.

[60] C. Cortes, M. Mohri, M. Riley, and A. Rostamizadeh. Sample selection bias correction theory. In *Proceedings of the 19th international conference on Algorithmic Learning Theory*, 2008.

[61] T. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman and Hall/CRC, 2000.

[62] K. Crowston and M. Williams. Reproduced and emergent genres of communication on the world-wide web. *The Information Society*, 16(3): 201–216, 2000.

[63] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu. Transferring naive Bayes classifiers for text classification. In *Proceedings of the 22nd national conference on Artificial intelligence*, 2007.

[64] E. Dale and J. Chall. A formula for predicting readability. *Educational Research Bulletin*, 27, 1948.

[65] S. Das and M. Chen. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In *Proceedings of the annual conference of the Asia Pacific Finance Association*, 2001.

[66] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, 2003.

[67] M. de Buenaga Rodriguez, J. Gomez-Hidalgo, and B. Diaz-Agudo. Using WordNet to complement training information in text categorization. In *Proceedings of the 2nd international conference on Recent Advances in Natural Language Processing*, 1997.

[68] F. de Jong, H. Rode, and D. Hiemstra. Temporal language models for the disclosure of historical text. In *Proceedings of the 16th international conference of the Association for History and Computing*, 2005.

[69] N. Dewdney, C. VanEss-Dykema, and R. MacMillan. The form is the substance: Classification of genres in text. In *Proceedings of the ACL workshop on HumanLanguage Technology and Knowledge Management*, 2001.

[70] T. Dietterich and E. Kong. Machine learning bias, statistical bias, and statistical variance of decision tree algorithms. Department of Computer Science, Oregon State University, 1995.

[71] M. Dimitrova, A. Finn, N. Kushmerick, and B. Smyth. Web genre visualization. In *Proceedings of the conference on Human Factors in Computing Systems*, 2002.

[72] L. Dini and G. Mazzini. Opinion classification through information extraction. In *Proceedings of the conference on Data Mining Methods and Databases for Engineering, Finance and Other Fields*, 2002.

[73] P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.

[74] A. Dries and U. Rückert. Adaptive concept drift detection. *Statistical Analysis and Data Mining*, 2:311–327, 2009.

[75] G. Druck, G. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st international ACM SIGIR conference on Research and development in information retrieval*, 2008.

[76] Y. Duan, L. Jiang, T. Qin, M. Zhou, and H.-Y. Shum. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd international conference on Computational Linguistics*, 2010.

[77] T. Dunning. Statistical identification of language. Computing Research Laboratory, New Mexico State, 1994.

[78] S. Ertekin, J. Huang, L. Bottou, and L. Giles. Learning on the border: active learning in imbalanced data classification. In *Proceedings of the 16th ACM conference on Information and knowledge management*, 2007.

[79] S. Ertekin, J. Huang, and C. L. Giles. Active learning for class imbalance problem. In *Proceedings of the 30th international ACM SIGIR conference on Research and development in information retrieval*, 2007.

[80] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, 1996.

[81] T. Fawcett. ROC graphs: Notes and practical considerations for researchers. HP Laboratories, 2004.

[82] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[83] E. Filatova. Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In *Proceedings of the 8th international conference on Language Resources and Evaluation*, 2012.

[84] R. Finkel, A. Zaslavsky, K. Monostori, and H. Schmidt. Signature extraction for overlap detection in documents. In *Proceedings of the 25th Australasian conference on Computer science*, 2002.

[85] A. Finn and N. Kushmerick. Learning to classify documents according to genre. In *Proceedings of the workshop on Computational Approaches to Style Analysis and Synthesis*, 2003.

[86] R. Flesch. A new readability yardstick. *Journal of Applied Psychology*, 32: 221–233, 1948.

[87] G. Forman. *Computational Methods of Feature Selection*, chapter Feature Selection for Text Classification. Chapman and Hall/CRC, 2007.

[88] G. Forman. Tackling concept drift by temporal inductive transfer. In *Proceedings of the 29th international ACM SIGIR conference on Research and development in information retrieval*, 2006.

[89] L. Freund, C. Clarke, and E. Toms. Towards genre classification for ir in the workplace. In *Proceedings of the 1st international conference on Information interaction in context*, 2006.

[90] J. Friedman. On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77, 1997.

[91] N. Fuhr, S. Hartmann, G. Knorz, G. Lustig, M. Schwantner, and K. Tzeras. AIR/X – a rule-based multistage indexing system for large subject fields. In *Proceedings of the Recherche d'Information Assistee par Ordinateur*, 1991.

[92] E. Gabrilovich and S. Markovitch. Feature generation for text categorization using world knowledge. In *Proceedings of the 19th international joint conference on Artificial Intelligence*, 2005.

[93] D. Gamberger, N. Lavrac, and S. Dzeroski. Noise elimination in inductive concept learning: A case study in medical diagnosis. In *Algorithmic Learning Theory*. Springer-Verlag, 1996.

[94] M. Ganapathibhotla and B. Liu. Mining opinions in comparative sentences. In *Proceedings of the 22nd international conference on Computational Linguistics*, 2008.

[95] A. Garcia-Fernandez, A. Ligozat, M. Dinarelli, and D. Bernhard. When was it written? Automatically determining publication dates. In *Proceedings of the 18th international conference on String processing and information retrieval*, 2011.

[96] R. W. Gibbs. Irony in talk among friends. *Metaphor and Symbol*, 15(1-2): 5–27, 2000.

[97] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the 25th Very Large Data Bases conference*, 1999.

[98] J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarran. Indexing with wordnet synsets can improve text retrieval. In *Proceedings of the workshop on Usage of WordNet in Natural Language Processing Systems*, 1998.

[99] T. Gottron and N. Lipka. A comparison of language identification approaches on short, query-style texts. In *Proceedings of the 32nd European conference on Information Retrieval Research*, 2010.

[100] N. Graham, G. Hirst, and B. Marthi. Segmenting a document by stylistic character. *Natural Language Engineering*, 11(4):397–415, 2005.

[101] R. Gunning. *The Technique of Clear Writing*. McGraw-Hill, 1952.

[102] X. Guo, Y. Yin, C. Dong, G. Yang, and G. Zhou. On the class imbalance problem. In *Proceedings of the 4th international conference on Natural Computation*, 2008.

[103] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[104] U. Hanani, B. Shapira, and P. Shoval. Information filtering: Overview of issues, research and systems. *User Modeling and User-Adapted Interaction*, 11(3):203–259, 2001.

[105] D. J. Hand and K. Yu. Idiot's Bayes—not so stupid after all? *International Statistical Review*, 69:385–398, 2001.

[106] M. Harpalani, M. Hart, S. Signh, R. Johnson, and Y. Choi. Language of vandalism: Improving Wikipedia vandalism detection via stylometric analysis. In *Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.

[107] J. A. Hartigan and M. A. Wong. Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society*, 28:100–108, 1979.

[108] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[109] V. Hatzivassiloglou and K. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the joint conference of the Association for computational linuistics and the European chapter of the Association for computational linuistics*, 1997.

[110] M. A. Hearst. Direction-based text interpretation as an information access refinement. In *Text-based intelligent systems*. L. Erlbaum Associates, 1992.

[111] K. Hempstalk, E. Frank, and I. Witten. One-class classification by combining density and class probability estimation. In *Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases: Part I*, 2008.

[112] M. Henzinger. Finding near-duplicate Web pages: A large-scale evaluation of algorithms. In *Proceedings of the 29th international ACM SIGIR conference on Research and development in information retrieval*, 2006.

[113] R. J. Hickey. Noise modelling and evaluating learning from examples. *Artificial Intelligence*, 82(1-2):157–179, 1996.

[114] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313:504–507, 2006.

[115] H. O. Hirschfeld and J. Wishart. A connection between correlation and contingency. *Mathematical Proceedings of The Cambridge Philosophical Society*, 31(4), 1935.

[116] T. K. Ho. Random decision forests. In *Proceedings of the 3rd international conference on Document Analysis and Recognition*, 1995.

[117] W. H. Ho and P. A. Watters. Statistical and structural approaches to filtering internet pornography. In *Proceedings of the IEEE international conference on Systems, Man & Cybernetics*, 2004.

[118] T. Hoad and J. Zobel. Methods for identifying versioned and plagiarised documents. *American Society for Information Science and Technology*, 54(3): 203–215, 2003.

[119] D. Holmes. The evolution of stylometry in humanities scholarship. *Literary and Linguistic Computing*, 13(3):111–117, 1998.

[120] J. Holmes. Women's language: A functional approach. *General Linguistics*, 24(3):149–178, 1984.

[121] A. Honore. Some simple measures of richness of vocabulary. *Association for Literary and Linguistic Computing Bulletin*, 7(2):172–177, 1979.

[122] M. Hu, E.-P. Lim, A. Sun, H. W. Lauw, and B. Vuong. Measuring article quality in Wikipedia: models and evaluation. In *Proceedings of the 16th ACM international conference on Information and knowledge management*, 2007.

[123] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD conference on Knowledge Discovery and Data Mining*, 2004.

[124] W. Hu, O. Wu, Z. Chen, Z. Fu, and S. Maybank. Recognition of pornographic web pages by classifying texts and images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1019–1034, 2007.

[125] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems 19*, 2007.

[126] R. Huang. *Automatic dialect classification: Advances for read and spontaneous speech, and printed text*. PhD thesis, University of Colorado at Boulder, 2006.

[127] S. Huang. An active learning method for mining time-changing data streams. In *Proceedings of the 2nd international symposium on Intelligent Information Technology Application*, 2008.

[128] G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. In *Proceedings of the 7th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, 2001.

[129] P. Indyk and R. Motwani. Approximate nearest neighbor—towards removing the curse of dimensionality. In *Proceedings of the 30th symposium on Theory of Computing*, 1998.

[130] N. C. Ingle. A language identification table. *The Incorporated Linguist*, 15 (4), 1976.

[131] R. Jakobson. Closing statements: Linguistics and poetics. In *Style In Language*. MIT Press, 1960.

[132] N. Jindal and B. Liu. Opinion spam and analysis. In *Proceedings of the conference on Web Search and Web Data Mining*, 2008.

[133] T. Joachims. A statistical learning model of text classification with support vector machines. In *Proceedings of the 24th international ACM SIGIR conference on Research and development in information retrieval*, 2001.

[134] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of the 10th European conference on Machine Learning*, 1998.

[135] G. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of the 11th international conference on Machine Learning*, 1994.

[136] P. Juola. Authorship attribution. *Foundations and Trends in Information Retrieval*, 1(3):233–334, 2006.

[137] J. Justeson and S. Katz. Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1 (1):9–27, 1995.

[138] N. Kanhabua and K. Nørvåg. Improving temporal language models for determining time of non-timestamped documents. In *Proceedings of the 12th European conference on Research and Advanced Technology for Digital Libraries*, 2008.

[139] J. Karlgren and D. Cutting. Recognizing text genres with simple metrics using discriminant analysis. In *Proceedings of the 15th international conference on Computational Linguistics*, 1994.

[140] S. P. Kasiviswanathan, P. Melville, A. Banerjee, and V. Sindhwani. Emerging topic detection using dictionary learning. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011.

[141] A. Kennedy and M. Shepherd. Automatic identification of home pages on the Web. In *Proceedings of the 38th annual Hawaii international conference on System Sciences*, 2005.

[142] R. P. Kern. Usefulness of readability formulas for achieving Army readability objectives research and state-of-the-art applied to the Army's problems. U.S. Army Research Institute for the Behavioral and Social Sciences, 1980.

[143] B. Kessler, G. Nunberg, and H. Schütze. Automatic detection of text genre. In *Proceedings of the 8th international conference on Computational Linguistics and the 35th annual meeting of the Association for Computational Linguistics*, 1997.

[144] S. S. Khan and M. G. Madden. A survey of recent trends in one class classification. In *Proceedings of the 20th Irish conference on Artificial intelligence and cognitive science*, 2010.

[145] S.-M. Kim and E. Hovy. Determining the sentiment of opinions. In *Proceedings of the international conference on Computational Linguistics*, 2004.

[146] Y. Kim. An efficient text filter for adult web documents. In *Proceedings of the 8th international conference on Advanced Communication Technology*, 2006.

[147] J. Kincaid, R. Fishburne, R. Rogers, and B. Chissom. Derivation of new readability formulas (Automated Readability Index, Fog Count and Flesch Reading Ease formula) for Navy enlisted personnel. Research Branch Report 8-75 Millington TN: Naval Technical Training US Naval Air Station, 1975.

[148] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:226–239, 1998.

[149] B. Kjell, A. W. Woods, and O. Frieder. Discrimination of authorship using visualization. *Information Processing and Management*, 30(1):141–150, 1994.

[150] J. Kleinberg. Two algorithms for nearest-neighbor search in high dimensions. In *Proceedings of the 29th annual ACM symposium on Theory of computing*, 1997.

[151] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[152] M. Koppel and J. Schler. Exploiting stylistic idiosyncrasies for authorship attribution. In *Proceedings of the IJCAI workshop on Computational Approaches to Style Analysis and Synthesis*, 2003.

[153] M. Koppel and J. Schler. Authorship verification as a one-class classification problem. In *Proceedings of the 21st international conference on Machine learning*, 2004.

[154] M. Koppel, N. Akiva, and I. Dagan. Feature instability as a criterion for selecting potential style markers: Special topic section on computational analysis of style. *Journal of the American Society for Information Science and Technology*, 57(11):1519–1525, 2006.

[155] M. Koppel, J. Schler, and E. Bonchek-Dokow. Measuring differentiability: Unmasking pseudonymous authors. *Journal of Machine Learning Research*, 8:1261–1276, 2007.

[156] M. Koppel, N. Akiva, E. Alshech, and K. Bar. Automatically classifying documents by ideological and organizational affiliation. In *Proceedings of the 2009 IEEE international conference on Intelligence and security informatics*, 2009.

[157] M. Koppel, J. Schler, and S. Argamon. Computational methods in authorship attribution. *Journal of the American Society for Information Science and Technology*, 60(1):9–26, 2009.

[158] R. Krovetz. Viewing morphology as an inference process. In *Proceedings of the 16th international ACM SIGIR conference on Research and development in information retrieval*, 1993.

[159] J. Kubica and A. Moore. Probabilistic noise identification and data cleaning. In *Proceedings of the 3rd IEEE international conference on Data Mining*, 2004.

[160] G. Kumaran and J. Allan. Text classification and named entities for new event detection. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 2004.

[161] K. Lang. NewsWeeder: learning to filter netnews. In *Proceedings of the 12th international conference on Machine learning*, 1995.

[162] L. Larkey. Automatic essay grading using text categorization techniques. In *Proceedings the 21st ACM SIGIR international conference on Research and development in information retrieval*, 1998.

[163] H. D. Lasswell. The structure and function of communication in society. In *Communication of Ideas*. Harper & Co, 1953.

[164] D. Lawrie, W. Croft, and A. Rosenberg. Finding topic words for hierarchical summarization. In *Proceedings of the 24th international ACM SIGIR conference on Research and development in information retrieval*, 2001.

[165] D. J. Lawrie and W. Croft. Generating hierarchical summaries for Web searches. In *Proceedings of the 26th international ACM SIGIR conference on Research and development in information retrieval*, 2003.

[166] J. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Springer-Verlag, 2007.

[167] Y. Lee and S. Myaeng. Text genre classification with genre-revealing and subject-revealing features. In *Proceedings of the 25th international ACM SIGIR conference on Research and development in information retrieval*, 2002.

[168] E. W. D. L. Lena Grothe and A. Nürnberger. A comparative study on language identification methods. In *Proceedings of the international conference on Language Resources and Evaluation*, 2008.

[169] D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, 1992.

[170] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th international ACM SIGIR conference on Research and development in information retrieval*, 1994.

[171] D. Lewis, Y. Yang, T. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5: 361–397, 2004.

[172] X.-L. Li, B. Liu, and S.-K. Ng. Negative training data can be harmful to text classification. In *Proceedings of the 2010 conference on Empirical Methods in Natural Language Processing*, 2010.

[173] C. Lim, K. Lee, and G. Kim. Automatic genre detection of Web documents. In *Proceedings of the 2nd international joint conference on Natural Language Processing*, 2005.

[174] P. Lindstrom, B. Mac Namee, and S. J. Delany. Drift detection using uncertainty distribution divergence. In *Proceedings of the 11th IEEE international conference on Data Mining workshops*, 2011.

[175] C. X. Ling and V. S. Sheng. *Cost-sensitive learning and the class imbalance problem*. Springer-Verlag, 2008.

[176] N. Lipka, B. Stein, and J. Shanahan. Estimating the expected effectiveness of text classification solutions under subclass distribution shifts. In *Proceedings of the 12th IEEE international conference on Data Mining*, 2012.

[177] N. Lipka and B. Stein. Identifying featured articles in Wikipedia: Writing style matters. In *Proceedings of the 19th international conference on World Wide Web*, 2010.

[178] N. Lipka and B. Stein. Classifying with co-stems: A new representation for information filtering. In *Proceedings of the 33rd European conference on Information Retrieval Research*, 2011.

[179] N. Lipka and B. Stein. Robust models in information retrieval. In *Proceedings of the 22nd international workshop on Database and Expert Systems Applications*, 2011.

[180] N. Lipka, B. Stein, and M. Anderka. Cluster-based one-class ensemble for classification problems in information retrieval. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 2012.

[181] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu. Building text classifiers using positive and unlabeled examples. In *Proceedings of the 3rd IEEE international conference on Data Mining*, 2003.

[182] B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web*, 2005.

[183] H. Liu and H. Motoda. On issues of instance selection. *Data Mining and Knowledge Discovery*, 6(2):115–130, 2002.

[184] H. Liu and H. Motoda, editors. *Computational Methods of Feature Selection*. Chapman and Hall/CRC, 2007.

[185] K. Liu and J. Zhao. Cross-domain sentiment classification using a two-stage method. In *Proceedings of the 18th ACM international conference on Information and knowledge management*, 2009.

[186] S. Liu, M. X. Zhou, S. Pan, W. Qian, W. Cai, and X. Lian. Interactive, topic-based visual text summarization and analysis. In *Proceedings of the 18th ACM international conference on Information and knowledge management*, 2009.

[187] X.-Y. Liu and Z.-H. Zhou. Towards cost-sensitive learning for real-world applications. In *Proceedings of the Pacific-Asia conference on Knowledge Discovery and Data Mining Workshops*, 2011.

[188] X.-Y. Liu, J. Wu, and Z.-H. Zhou. Exploratory under-sampling for class-imbalance learning. In *Proceedings of the 6th IEEE international conference on Data Mining*, 2006.

[189] J. B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31, 1968.

[190] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. A survey of multilinear subspace learning for tensor data. *Pattern Recognition*, 44(7):1540–1551, 2011.

[191] C. Mallows. Some comments on C_P. *Technometrics*, 15(4):661–675, 1973.

[192] L. Manevitz and M. Yousef. One-class SVMs for document classification. *Journal of Machine Learning Research*, 2:139–154, 2001.

[193] J. Mansfield. Textbook plagiarism in PSY101 general psychology: incidence and prevention. In *Proceedings of the 18th annual conference on Undergraduate teaching of psychology: ideas and innovations*, 2004.

[194] T. Marill. On the effectiveness of receptors in recognition systems. *IEEE Transactions on Information Theory*, 9(1):11–17, 1963.

[195] M. E. Maron and J. L. Kuhns. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM*, 7:216–244, 1960.

[196] M. Maron. Automatic indexing: an experimental inquiry. *Journal of the ACM*, 8:404–417, 1961.

[197] O. Mazhelis. One-class classifiers : a review and analysis of suitability in the context of mobile-masquerader detection. *South African Computer Journal*, 36:29–48, 2006.

[198] D. L. McGuinness, H. Zeng, P. P. da Silva, L. Ding, D. Narayanan, and M. Bhaowal. Investigations into trust for collaborative information repositories: A Wikipedia case study. In *Proceedings of the 15th International World Wide Web conference*, 2006.

[199] T. C. Mendenhall. The characteristic curves of composition. *Science*, 11: 237–49, 1887.

[200] T. C. Mendenhall. A mechanical solution of a literary problem. *Mendenhall*, LX(60):97–105, 1901.

[201] S. Meyer zu Eißen and B. Stein. Genre classification of Web pages: User study and feasibility analysis. In *Advances in Artificial Intelligence*, 2004.

[202] S. Meyer zu Eißen and B. Stein. Intrinsic plagiarism detection. In *Proceedings of the 28th European conference on Information Retrieval Research*, 2006.

[203] S. Meyer zu Eißen, B. Stein, and M. Kulig. Plagiarism detection without reference collections. In *Advances in Data Analysis*, 2007.

[204] R. Mihalcea and C. Strapparava. Learning to laugh (automatically): Computational models for humor recognition. *Computational Intelligence*, 22(2): 126–142, 2006.

[205] A. Minamikawa and H. Yokoyama. Personality estimation based on weblog text classification. In *Proceedings of the 24th international conference on Industrial engineering and other applications of applied intelligent systems conference on Modern approaches in applied intelligence*, 2011.

[206] T. Mitchell. *Machine Learning*. McGraw-Hill Higher Education, 1997.

[207] D. Mladenic. *Machine learning on non-homogeneous, distributed text data*. PhD thesis, University of Ljubljana, 1998.

[208] D. Mladenic and M. Grobelnik. Word sequences as features in text learning. In *Proceedings of the 17th Electrotechnical and Computer Science conference*, 1998.

[209] C. N. Mooers. Information retrieval viewed as temporal signaling. In *Proceedings of the international congress of Mathematicians*, 1950.

[210] C. N. Mooers. The next twenty years in information retrieval: some goals and predictions. In *Proceedings of the Western joint Computer conference*, 1959.

[211] B. M. E. Moret. Decision trees and diagrams. *ACM Computing Surveys*, 14 (4):593–623, 1982.

[212] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima. Mining product reputations on the Web. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2002.

[213] F. Mosteller and D. Wallace. *Inference and Disputed Authorship: Federalist Papers*. Addison-Wesley, 1964.

[214] P. M. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers*, 26(9):917–922, 1977.

[215] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems 14*, 2002.

[216] K. Nguyen. Classification of corporate and public text. CS229, Stanford University, 2011.

[217] M. N. Nguyen, X. Li, and S.-K. Ng. Ensemble based positive unlabeled learning for time series classification. In *Proceedings of the 17th international conference on Database systems for advanced applications*, 2012.

[218] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam Web pages through content analysis. In *Proceedings of the 15th international conference on World Wide Web*, 2006.

[219] M. Oakes and Y. Xu. A search engine based on query logs, and search log analysis at the university of Sunderland. In *Proceedings of the 10th Cross Language Evaluation Forum*, 2009.

[220] N. Oza and K. Tumer. Classifier ensembles: Select real-world applications. *Information Fusion*, 9(1):4–20, 2008.

[221] C. D. Paice. Another stemmer. *SIGIR Forum*, 24(3):56–61, 1990.

[222] B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, 2004.

[223] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[224] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, 2002.

[225] P. Papadimitriou and H. Garcia-Molina. Data leakage detection. *IEEE Transactions on Knowledge and Data Engineering*, 23:51–63, 2011.

[226] K. Pearson. On lines and planes of closest fit to a system of points in space. *Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2 (6):559–572, 1901.

[227] J. W. Pennebaker and L. D. Stone. Words of wisdom: Language use over the life span. *Journal of Personality and Social Psychology*, 85(2):291–301, 2003.

[228] J. Pennebaker, M. Mehl, and K. Niederhoffer. Psychological aspects of natural language use: Our words, our selves. *Annual Reviews of Psychology*, 54:547–577, 2003.

[229] R. Perdisci, G. Gu, and W. Lee. Using an ensemble of one-class SVM classifiers to harden payload-based anomaly detection systems. In *Proceedings of the 6th IEEE international conference on Data Mining*, 2006.

[230] J. C. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. *Advances in Large Margin Classifiers*, pages 61–74, 2000.

[231] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the Human Language Technology conference and the conference on Empirical Methods in Natural Language Processing*, 2005.

[232] A. Popescul and L. Ungar. Automatic labeling of document clusters. University of Pennsylvania, 2000.

[233] M. F. Porter. An algorithm for suffix stripping. *Program: Electronic Library & Information Systems*, 40(3):211–218, 1980.

[234] M. Potthast, B. Stein, and M. Anderka. A Wikipedia-based multilingual retrieval model. In *Proceedings of the 30th European conference on Information Retrieval Research*, 2008.

[235] M. Potthast, B. Stein, and R. Gerling. Automatic vandalism detection in Wikipedia. In *Proceedings of the 30th European conference on Information Retrieval Research*, 2008.

[236] P. Prettenhofer and B. Stein. Cross-lingual adaptation using structural correspondence learning. *ACM Transactions on Intelligent Systems and Technology*, 3:13:1–13:22, 2011.

[237] R. Priedhorsky, J. Chen, S. T. K. Lam, K. Panciera, L. Terveen, and J. Riedl. Creating, destroying, and restoring value in Wikipedia. In *Proceedings of the international ACM conference on Supporting Group Work*, 2007.

[238] L. Qiu, W. Zhang, C. Hu, and K. Zhao. SELC: a self-supervised model for sentiment classification. In *Proceedings of the 18th ACM international conference on Information and knowledge management*, 2009.

[239] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[240] G. Rätsch, S. Mika, B. Schölkopf, and K.-R. Müller. Constructing boosting algorithms from SVMs: An application to one-class classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1184–1199, 2002.

[241] A. Rauber and A. Müller-Kögler. Integrating automatic genre analysis into digital libraries. In *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, 2001.

[242] N. M. Razali and Y. B. Wah. Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests. *Journal of Statistical Modeling and Analytics*, 2:21–33, 2011.

[243] J. Read. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research workshop*, 2005.

[244] G. Rehm. Towards automatic Web genre identification. In *Proceedings of the 35th annual Hawaii international conference on System Sciences*, 2002.

[245] R. Rehurek and M. Kolkus. Language identification on the Web: Extending the dictionary method. In *Proceedings of the 10th international conference on Computational Linguistics and Intelligent Text Processing*, 2009.

[246] A. Reyes, P. Rosso, and D. Buscaldi. From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering*, 74:1–12, 2012.

[247] J. Reynar. *Topic Segmentation: Algorithms and Applications*. PhD thesis, University of Pennsylvania, 1998.

[248] E. Riloff. Little words can make a big difference for text classification. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, 1995.

[249] S. E. Robertson. The probability ranking principle in IR. *Journal of Documentation*, 33(4):294–304, 1977.

[250] D. Roussinov, K. Crowston, M. Nilan, B. Kwasnik, J. Cai, and X. Liu. Genre based navigation on the Web. In *Proceedings of the 34th Hawaii international conference on System Sciences*, 2001.

[251] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the 18th international conference on Machine learning*, 2001.

[252] S. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice-Hall, 1995.

[253] W. Sack. On the computation of point of view. In *Proceedings of AAAI*, 1994.

[254] G. Salton and M. E. Lesk. The SMART automatic document retrieval system—an illustration. *Communications of the ACM*, 8(6):391–398, 1965.

[255] G. Salton, A. Wong, and C. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[256] C. Sanderson and S. Guenter. On authorship attribution via Markov chains and sequence kernels. In *Proceedings of the 18th international conference on Pattern Recognition*, 2006.

[257] M. Santini. Common criteria for genre classification: Annotation and granularity. In *Proceedings of the 3rd international workshop on Text-Based Information Retrieval*, 2006.

[258] M. Santini. *Automatic Identification of Genre in Web Pages*. PhD thesis, University of Brighton, 2007.

[259] I. Santos, P. Galán-García, A. Santamaría-Ibirika, B. Alonso-Isla, I. Alabau-Sarasola, and P. Bringas. Adult content filtering through compression-based text classification. In *Special sessions of the international joint conference CISIS-ICEUTE-SOCO*. Springer-Verlag, 2013.

[260] C. Schaffer. Overfitting avoidance as bias. *Machine Learning*, 10(2):153–178, 1993.

[261] C. Schaffer. A conservation law for generalization performance. In *Proceedings of the 11th international conference on Machine learning*, 1994.

[262] J. Schler, M. Koppel, S. Argamon, and J. Pennebaker. Effects of age and gender on blogging. In *Proceedings of the AAAI symposium on Computational Approaches for Analyzing Weblogs*, 2006.

[263] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 2001.

[264] F. Schulz von Thun. *Miteinander Reden. 1: Störungen und Klärungen*. Reinbek bei Hamburg, 1981.

[265] S. Scott and S. Matwin. Feature engineering for text classification. In *Proceedings of the 16th international conference on Machine Learning*, 1999.

[266] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34, 2002.

[267] B. Settles, M. Craven, and S. Ray. Multiple instance active learning. In *Advances in Neural Information Processing Systems 20*, 2008.

[268] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the 5th annual workshop on Computational learning theory*, 1992.

[269] A. Shabtai, Y. Elovici, and L. Rokach. *A Survey of Data Leakage Detection and Prevention Solutions*. Springer-Verlag, 2012.

[270] J. Shanahan, N. Lipka, and D. Van den Poel. Learning to active learn with applications in the online advertising field of look-alike modeling. In *Online proceedings of the Direct/Interactive Marketing Research summit*, 2011.

[271] J. G. Shanahan and N. Roma. Boosting support vector machines for text classification through parameter-free threshold relaxation. In *Proceedings of the 12th ACM international conference on Information and knowledge management*, 2003.

[272] A. Shieh and D. Kamm. Ensembles of one class support vector machines. In *Multiple Classifier Systems*. Springer-Verlag, 2009.

[273] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227–244, 2000.

[274] P. Sibun and J. C. Reynar. Language identification: Examining the issues, 1996.

[275] Y. L. Simmhan, B. Plale, and D. Gannon. A survey of data provenance in e-science. *ACM SIGMOD Record*, 34(3):31–36, 2005.

[276] V. Sindhwani and S. Keerthi. Large scale semi-supervised linear SVMs. In *Proceedings of the 29th international ACM SIGIR conference on Research and development in information retrieval*, 2006.

[277] S. Singhi and H. Liu. Feature subset selection bias for classification learning. In *Proceedings of the 23rd international conference on Machine learning*, 2006.

[278] N. Slonim and N. Tishby. The power of word clusters for text classification. In *Proceedings of the 23rd european colloquium on Information Retrieval Research*, 2001.

[279] C. Spearman. General intelligence, objectively determined and measured. *American Journal of Psychology*, 15:201–293, 1904.

[280] K. Sridharan and S. Kakade. An information theoretic framework for multi-view learning. In *Proceedings of the 19th annual conference on Computational Learning Theory*, 2008.

[281] E. Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60, 2009.

[282] E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Text genre detection using common word frequencies. In *Proceedings of the 18th international conference on Computational Linguistics*, 2000.

[283] E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Computer-based authorship attribution without lexical measures. *Computers and the Humanities*, 35:193–214, 2001.

[284] E. Stamatatos. Author identification using imbalanced and limited training texts. In *Proceedings of the 18th international conference on Database and Expert Systems Applications*, 2007.

[285] M. Stefik. *Introduction to Knowledge Systems*. Morgan Kaufmann, 1995.

[286] J. Steiger. Factor indeterminacy in the 1930's and the 1970's some interesting parallels. *Psychometrika*, 44:157–167, 1979.

[287] B. Stein. Fuzzy-fingerprints for text-based information retrieval. In *Proceedings of the 5th international conference on Knowledge Management*, 2005.

[288] B. Stein. Principles of hash-based text retrieval. In *Proceedings of the 30th international ACM SIGIR conference on Research and development in information retrieval*, 2007.

[289] B. Stein and M. Busch. Density-based cluster algorithms in low-dimensional and high-dimensional applications. In *Proceedings of the 2nd international workshop on Text-Based Information Retrieval*, 2005.

[290] B. Stein and S. Meyer zu Eißen. Topic identification: Framework and application. In *Proceedings of the 4th international conference on Knowledge Management*, 2004.

[291] B. Stein and S. Meyer zu Eißen. Topic-Identifikation: Formalisierung, Analyse und neue Verfahren. *Künstliche Intelligenz*, 3:16–22, 2007.

[292] B. Stein and S. Meyer zu Eißen. Retrieval models for genre classification. *Scandinavian Journal of Information Systems*, 20:91–117, 2008.

[293] B. Stein, N. Lipka, and S. Meyer zu Eißen. Meta analysis within authorship verification. In *Proceedings of the 19th international workshop on Database and Expert Systems Applications*, 2008.

[294] B. Stein, N. Lipka, and P. Prettenhofer. Intrinsic plagiarism analysis. *Language Resources and Evaluation*, 45:63–82, 2011.

[295] B. Stein, S. Meyer zu Eißen, and N. Lipka. Web genre analysis: Use cases, retrieval models, and implementation issues. In *Genres on the Web*. Springer-Verlag, 2011.

[296] K. Stein and C. Hess. Does it matter who contributes: a study on featured articles in the german Wikipedia. In *Proceedings of the 18th conference on Hypertext and hypermedia*, 2007.

[297] G. W. Stewart. On the early history of the singular value decomposition. *SIAM Review*, 35(4):551–566, 1993.

[298] B. Stvilia, M. B. Twidale, L. C. Smith, and L. Gasser. Assessing information quality of a community-based encyclopedia. In *Proceedings of the international conference on Information Quality*, 2005.

[299] C. Y. Suen. N-gram statistics for natural language understanding and text processing. *IEEE Transactions on Pattern Analysis and Machine Ingelligence*, PAMI-1(2):164–172, 1979.

[300] R. Swan and D. Jensen. Timemines: Constructing timelines with statistical models of word usage. In *Proceedings of the 6th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, 2000.

[301] M. Taboga. *Lectures on Probability Theory and Mathematical Statistics*. CreateSpace Independent Publishing Platform, 2012.

[302] K. Tateishi, Y. Ishiguro, and T. Fukushima. Opinion information retrieval from the Internet. *Information Processing Society of Japan*, pages 75–82, 2001.

[303] D. Tax. *One-Class Classification*. PhD thesis, Delft University of Technology, 2001.

[304] D. Tax and R. Duin. Combining one-class classifiers. In *Proceedings of the 2nd international workshop on Multiple Classifier Systems*, 2001.

[305] W. J. Teahan. Text classification and segmentation using minimum cross-entropy. In *Proceedings of the Recherche d'Information Assistee par Ordinateur*, 2000.

[306] C.-M. Teng. Correcting noisy data. In *Proceedings of the 16th international conference on Machine Learning*, 1999.

[307] R. M. Tong. An operational system for detecting and tracking opinions in on-line discussion. In *Proceedings of the workshop on Operational text classification*, 2001.

[308] O. Tsur, D. Davidov, and A. Rappoport. A great catchy name: Semi-supervised recognition of sarcastic sentences in product reviews. In *Proceedings of the 4th international conference on Weblogs and Social Media*, 2010.

[309] P. Turney. Technical note: Bias and the quantification of stability. *Machine Learning*, 20(1-2):23–33, 1995.

[310] F. Tweedie and H. Baayen. How variable may a constant be? Measures of lexical richness in perspective. *Computers and the Humanities*, 32(5):323–352, 1998.

[311] L. Ureña-López, M. Buenaga, and J. Gómez. Integrating linguistic resources in TC through WSD. *Computers and the Humanities*, 35:215–230, 2001.

[312] H. Vafai and K. De Jong. Genetic algorithms as a tool for feature selection in machine learning. In *Proceedings of the 4th international conference on Tools with artificial intelligence*, 1992.

[313] G. Valentini and T. Dietterich. Bias-variance analysis of support vector machines for the development of SVM-based ensemble methods. *The Journal of Machine Learning Research*, 5:725–775, 2004.

[314] H. van Halteren. Author verification by linguistic profiling: An exploration of the parameter space. *ACM Transactions on Speech and Language Processing*, 4(1):1:1–1:17, 2007.

[315] H. van Halteren and N. Oostdijk. Linguistic profiling of texts for the purpose of language verification. In *Proceedings of the 20th international conference on Computational Linguistics*, 2004.

[316] C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979.

[317] C. J. van Rijsbergen. Towards an information logic. *SIGIR Forum*, 23(SI): 77–86, 1989.

[318] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 2000.

[319] T. Vatanen, J. J. Väyrynen, and S. Virpioja. Language identification of short text segments with n-gram models. In *Proceedings of the international conference on Language Resources and Evaluation*, 2010.

[320] P. Vojtek and M. Bieliková. Comparing natural language identification methods based on Markov processes. In *Proceedings of the 4th international seminar Computer Treatment of Slavic and East European Languages*, 2007.

[321] J. Vreeken, M. van Leeuwen, and A. Siebes. Characterising the difference. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, 2007.

[322] B. C. Wallace, K. Small, C. E. Brodley, and T. A. Trikalinos. Class imbalance, redux. In *Proceedings of the 11th IEEE international conference on Data Mining*, 2011.

[323] B. B. Wang, R. I. B. Mckay, H. A. Abbass, and M. Barlow. A comparative study for domain ontology guided feature extraction. In *Proceedings of the 26th Australasian computer science conference*, 2003.

[324] D. Wang, D. S. Yeung, and E. C. C. Tsang. Structured one-class classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 36: 1283–1295, 2006.

[325] R. Y. Wang, V. C. Storey, and C. P. Firth. A framework for analysis of data quality research. *IEEE Transactions on Knowledge and Data Engineering*, 7(4): 623–640, 1995.

[326] W. Y. Wang and K. McKeown. "Got You!": Automatic vandalism detection in Wikipedia with Web-based shallow syntactic-semantic modeling. In *Proceedings of the 23rd international conference on Computational Linguistics*, 2010.

[327] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang. Topic sentiment analysis in Twitter: a graph-based hashtag sentiment classification approach. In *Proceedings of the 20th ACM international conference on information and knowledge management*, 2011.

[328] J. H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58:236–244, 1963.

[329] K. Q. Weinberger and L. K. Saul. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *Proceedings of the 21st national conference on Artificial intelligence - Volume 2*, 2006.

[330] J. Wiebe. Learning subjective adjectives from corpora. In *Proceedings of AAAI*, 2000.

[331] J. M. Wiebe. Identifying subjective characters in narrative. In *Proceedings of the international conference on Computational Linguistics*, 1990.

[332] J. M. Wiebe and W. J. Rapaport. A computational theory of perspective and reference in narrative. In *Proceedings of the Association for Computational Linguistics*, 1988.

[333] Y. Wilks and J. Bien. Beliefs, points of view and multiple environments. In *Proceedings of the international NATO symposium on Artificial and human intelligence*, 1984.

[334] C. B. Williams. Mendenhall's studies of word-length distribution in the works of Shakespeare and Bacon. *Biometrika*, 62:207–212, 1975.

[335] D. R. Wilson and T. R. Martinez. Bias and the probability of generalization. In *Proceedings of the international conference on Intelligent Information Systems*, 1997.

[336] D. L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man and Cybernetics*, 2(3):408–421, 1972.

[337] S. K. M. Wong, W. Ziarko, and P. C. N. Wong. Generalized vector spaces model in information retrieval. In *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, 1985.

[338] D. Xing, W. Dai, G.-R. Xue, and Y. Yu. Bridged refinement for transfer learning. In *Proceedings of the 11th European conference on Principles and Practice of Knowledge Discovery in Databases*, 2007.

[339] G. Xue, D. Xing, Q. Yang, and Y. Yu. Deep classification in large-scale text hierarchies. In *Proceedings of the 31st international ACM SIGIR conference on Research and development in information retrieval*, 2008.

[340] C. Yang, K. H.-Y. Lin, and H.-H. Chen. Emotion classification using Web blog corpora. In *Proceedings of the IEEE/WIC/ACM international conference on Web Intelligence*, 2007.

[341] H. Yang and J. Callan. Near-duplicate detection by instance-level constrained clustering. In *Proceedings of the 29th international ACM SIGIR conference on Research and development in information retrieval*, 2006.

[342] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *Proceedings of the 4th ACM international conference on Web search and data mining*, 2011.

[343] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22th international ACM SIGIR conference on Research and development in information retrieval*, 1999.

[344] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the 14th international conference on Machine Learning*, 1997.

[345] B. Yazici and S. Yolacan. A comparison of various tests of normality. *Journal of Statistical Computation and Simulation*, 77:175–183, 2007.

[346] T. Yoshioka and G. Herman. Coordinating information using genres. Massachusetts Institute of Technology (MIT), Sloan School of Management, 2000.

[347] B. Yu. An evaluation of text classification methods for literary study. *Literary & Linguistic Computing*, 23(3):327–343, 2008.

[348] G. U. Yule. On sentence-length as a statistical characteristic of style in prose: With application to two cases of disputed authorship. *Biometrika*, 30:363–390, 1939.

[349] G. U. Yule. *The Statistical Study of Literary Vocabulary*. Cambridge University Press, 1944.

[350] S. Zelikovitz and H. Hirsh. Using LSI for text classification in the presence of background text. In *Proceedings of the 10th international conference on Information and knowledge management*, 2001.

[351] H. Zeng, M. A. Alhossaini, L. Ding, R. Fikes, and D. L. McGuinness. Computing trust from revision history. In *Proceedings of the conference on Privacy, Security and Trust*, 2006.

[352] Y. Zhang, X. Li, and M. E. Orlowska. One-class classification of text streams with concept drift. In *Proceedings of the 8th IEEE international conference on Data Mining workshops*, 2008.

[353] R. Zheng, J. Li, H. Chen, and Z. Huang. A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American Society for Information Science and Technology*, 57:378–393, 2006.

[354] X. Zhu and X. Wu. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22(3):177–210, 2003.

[355] X. Zhu, X. Wu, and Q. Chen. Eliminating class noise in large datasets. In *Proceedings of the 20th international conference on Machine Learning*, 2003.

[356] X. Zhu, X. Wu, and Y. Yang. Error detection and impact-sensitive instance ranking in noisy datasets. In *Proceedings of the 19th national conference on Artifical intelligence*, 2004.

[357] X. Zhu, X. Wu, T. M. Khoshgoftaar, and Y. Shi. An empirical study of the noise impact on cost-sensitive learning. In *Proceedings of the 20th international joint conference on Artificial Intelligence*, 2007.

[358] G. K. Zipf. *Selective Studies and the Principle of Relative Frequency in Language*. Harvard University Press, 1932.