Combining Text and Linguistic Document Representations for Authorship Attribution

Andreas Kaster, Stefan Siersdorfer, Gerhard Weikum
{kaster,stesi,weikum}@mpi-sb.mpg.de

Max-Planck-Institute for Computer Science, Germany

ABSTRACT

In this paper, we provide several alternatives to the classical Bag-Of-Words model for automatic authorship attribution. To this end, we consider linguistic and writing style information such as grammatical structures to construct different document representations. Furthermore we describe two techniques to combine the obtained representations: combination vectors and ensemble based meta classification. Our experiments show the viability of our approach.

General Terms

text classification, authorship attribution

Keywords

linguistic information, writing style information, combination techniques, stylometry

1. INTRODUCTION

1.1 Motivation

Automatic document classification is useful for a wide range of applications such as organizing Web, intranet or portal pages into topic directories, filtering news feeds or mail, focused crawling on the Web or in intranets and many more [11].

Most of the text classification approaches deal with topic-oriented classification (e.g., classifying documents into classes like "Sports", "Politics" or "Computer Science"). Here the Bag-Of-Words model, taking just the occurences of words into account (often using additional techniques like stemming, stopword elimination, and different weighting schemes), has been shown to be very effective for this task [20, 45].

But these techniques have limitations for other classification tasks such as authorship recognition. In this context, application scenarios include tasks like plagiarism detection [22], author identification forensics [12], author tracking in discussion forums [31], or solving problems of disputed authorship for historic documents such as the Federalist problem [18, 29].

Although in the case of authorship attribution, there occurs also a certain amount of topic and word correlation (in books written by Doyle we will typically find the names "Holmes" and "Watson", in books written by Christie we have, e.g., "Poirot" and "Marple"), alternative features (e.g., features, that do not contain any information about a document's content at all) may become important.

In this paper we study, in addition to some known approaches (like function words, filtering based on Part-Of-Speech tagging), several new approaches for feature construction. These include writing style features using syntax trees, considering constituents, and statistical measures on tree depths. As a result we obtain different and, to a certain degree, orthogonal document representations.

We combine these representations using two different techniques: combination vectors and meta classification.

1.2 Contribution

The paper makes the following contributions:

- In addition to using some known techniques we describe several novel approaches for the construction of document features for authorship attribution.
- We describe two different ways to combine and weight distinct feature spaces.
- We provide an experimental study of the pros and cons of a variety of methods.

1.3 Outline

The rest of the paper is organized as follows. In Section 2 we briefly review the technical basics of automatic classification and some linguistic basics. In Section 3 we describe different feature representations of documents. Combination methods for different document representations are described in Section 4. Section 5 presents experiments on a dataset based on Project Gutenberg [1]. Finally, in Section 6 we discuss related work in comparison to our own research.

2. TECHNICAL BASICS

2.1 Machine Learning

Classifying text documents into thematic categories usually follows a supervised learning paradigm and is based on training documents that need to be provided for each topic. Both training documents and test documents, which are later given to the classifier, are represented as multi-dimensional feature vectors. In the prevalent bag-of-words model the features are derived from word occurrence frequencies, e.g. based on tf*idf feature weights [6, 26]. Often feature selection algorithms are applied to reduce the dimensionality of the feature space and eliminate "noisy", non-characteristic features, based on information-theoretic measures for feature ordering (e.g., relative entropy or information gain).

The resulting compact feature vectors are used to derive a classification model for each topic, using probabilistic (e.g., Naive Bayes) or discriminative models (e.g., SVM). Linear support vector machines (SVMs) construct a hyperplane $\vec{w} \cdot \vec{x} + b = 0$ that separates the set of positive training examples from a set of negative examples with maximum margin. This training requires solving a quadratic optimization problem whose empirical performance is somewhere between quadratic and cubic in the number of training documents and linear in the number of features [9]. For a new, previously unseen, document $\vec{d}$ the SVM merely needs to test whether the document lies on the "positive" side or the "negative" side of the separating hyperplane. The decision simply requires computing a scalar product of the vectors $\vec{w}$ and $\vec{d}$. SVMs have been shown to perform very well for text classification (see, e.g., [14, 19]).

2.2 Linguistic Basics

For the understanding of techniques described below, we introduce some basic concepts. Further details will be provided later when needed. Consider a set Σ of tags for linguisitic corpus annotation (e.g. the Penn-Treebank-Tagset [27]). Let s be a sentence and $T_s := (V, E, \sigma)$ an ordered tree with a set of nodes V, a set of edges E and a labeling function $\sigma : V \rightarrow \Sigma$, that assigs a label $l \in \Sigma$ to each node of the tree. We call T_s the syntax tree of sentence s. T_s is the tree representation of a probabilistic contextfree grammar (PCFG). A PCFG is a contextfree grammar enriched by transition probabilities for each rewriting rule ([26]). For example, consider Figure 1. There, the sentence *Next, he examined the framework of the door we had broken in, assuring himself that the bolt had really been shot.* is represented as a syntax tree. The leaves of the tree represent the words themselves, i.e. terminal symbols, where the higher nodes represent the PCFG Tags, i.e., non terminal symbols. Non-terminals can be subdevided into other non-terminals or terminals, e.g. NP (a noun phrase) into DT (determiner, an article) and NN (a noun in singular case) and NN into "framework". Intuitively, a syntax tree represents the structure of a sentence, and, in some way, the writing style of an author, which we use for feature construction as described in Section 3.4.

3. CONSTRUCTION OF FEATURES

3.1 Word-Based Features

Using word based features is the most popular and, despite of its simplicity, very effective feature construction method. We briefly describe several variants from the literature, that we will consider as a baseline for other methods.

3.1.1 Bag-Of-Words

In the Bag-Of-Words approach the ordering of the words is not considered. Optionally a stopword list can be used to eliminate very common terms like articles, prepositions, etc. Often additional techniques like stemming [35] are applied to the words. There are different options to construct feature weights: taking the absolute or relative frequency of term occurrences as compontents, constructing a binary feature vector by just considering the pure occurrence of a term, computing the tf*idf values of the terms, etc. [30, 37]. Because it is the state-of-the-art method for feature construction in automatic document classification we will consider Bag-Of-Words as baseline for our experiments.

3.1.2 Function Words

In case of authorship attribution it can make sense to use "content-free" features, i.e., terms, that do not contain information about the document's content such as prepositions, pronouns, determiners etc. These terms are called *function words* (see e.g. Diederich et. al. [13]). In our implementation we regard as function words all words other than nouns, verbs and adjectives.

3.1.3 POS Annotation

In this approach, the part of speech (POS) group of the words (e.g. verb, noun, adjective) is taken into account [33]. This can be used to filter documents, e.g., by considering only nouns or verbs. POS is also used for simple disambiguation, e.g., by distinguishing the verb "book" from the noun "book".

3.1.4 Feature Selection

The idea of feature selection is to just take the most discriminating features into account. Intuitively a well discriminating term for two classes A and B occurs frequently in documents of class A and infrequently in documents of class B or vice versa. Examples of feature selection measures are Mutual Information, Information Gain, and Chi Square [46].

3.1.5 Semantic Disambiguation

Here a thesaurus, e.g. Wordnet [15], is used to disambiguate terms (treating synonyms like "automobile" and "car" as the same feature). In some approaches also more complex relationships between words are taken into account [36, 38].

3.2 Using Linguistic Constituents

The structure of natural language sentences shows that word occurrences follow a specific order, called word order. Words are grouped into syntactic units, *constituents*, that can be deeply nested. Such constituents can be detected by their being able to occur in various positions and showing uniform syntactic possibilities for expansion (see [26]). Consider again the sentence *Next, he examined the framework of the door we had broken in, assuring himself that the bolt had really been shot.* and its syntax tree representation in Figure 1. In particular, consider the part *he examined the framework*. This part is a constituent of the sentence with sub-constituents, e.g. *"the framework"*. The sub-constituents can change their positions inside the bigger constituent. Just considering that specific part, *he examined the framework* has the same meaning as *the framework he examined*. We can use this information about the word relationships by extracting constituents for feature construction. To this end, we first subdivide the document into sentences, and then construct a syntax tree as shown in Figure 1 for each sentence (note that the grey-boxed parts belong to another technique, the writing style, described in 3.4). In our framework, we use the Connexor Machinese Phrase Tagger [41] to subdivide a document into sentences and Lexparser [2] to build the syntax trees. We define a minimal and maximal length, min and max, of the constituents that we want to use for feature construction.

The simplest way to construct features would be to just concatenate the features inside a constituent using an appropriate separation character (e.g. "$").
From our example sentence, this would result features such as $he\$examined\$the\$framework$. But such very specific features may occur very rarely in the document corpus. To obtain more "common" features a combination of some of the following options is applied:

- Performing stemming and stopword elimination on the words contained in a constituent.
- Abstracting from the ordering of words by putting the words into lexicographic order.
- Instead of a feature $x_1\$x_2\$\ldots\$x_n$ consider pairs $x_i\$x_j$ or triples contained in the constituents (bi- and trigrams).
- Perform a feature selection on the constituent-features themselves. This can be done completely analogously to the feature selection for simple words.

In Section 5, we provide experiments on using whole constituents with stemming and stopword-elimination as well as using bigrams (also stemmed and stopword-cleaned).

3.3 Functional Dependencies

Functional dependencies represent relational information in sentences. Consider again a part of the sentence used in Figure 1, *he examined the framework of the door*. Here, *he* is the subject (agent) and *framework* and *door* are the objects of the predicate *examined* (action). We used the Connexor Machinese Syntax [41] to determine such dependencies. Our features have the form

$$x_1\$x_2\$\ldots\$x_n \tag{1}$$

where x_1 is the subject of an action, x_2 is the predicate and x_3 through x_n are the objects. To obtain a canonical form, words are reduced to their base forms, using Connexor Machinese Phrase Tagger [41], and objects are sorted in lexicographic order. In our example case, we get the feature $he\$examine\$door\$framework$.

3.4 Writing Style: Using Syntax Trees

Different authors may construct sentences in their writings in a completely different way. The idea is to consider a syntax tree representation of their sentences as features. In the extreme case we could encode the whole tree into a string; but this would result in very sparse feature spaces. Instead we should restrict ourselves to nodes up to a certain maximum tree depth. In our experiments we observed that considering just the children of the root nodes of sentences and sub-clauses (labeled with S) provides us already with interesting features. So our example tree in Figure 1 could be encoded into the features $ADVP\$, \$NP\$VP$, VP, and two times $NP\$VP$ (emphasized by the grey boxes). Note that this method does not use any word information at all. Table 1 shows the top-5 features for the authors A. C. Doyle and R. Burton according to their mutual information values (we considered only books available from the Gutenberg Project [1]; see Section 5 for details). We do not apply any kind of filtering mechanism to the structure features such as removing punctuation marks. Experiments showed that those marks provide interesting information about the sentence structure. Note, that "," in the feature $ADVP\$, \$NP\$VP$ represents an annotation tag for a word phrase in the syntax tree, not the comma itself.

A.C. Doyle		R. Burton	
Feature	**MI**	**Feature**	**MI**
S$,$CCS.	0.23	S$:$S	0.26
PPNPVP$.	0.16	SCCS	0.23
SBAR$,$NPVP.	0.14	XXNP$VP	0.21
SBAR$,$XNPVP$.	0.13	S$:$S$.	0.20
PP$,$NPVP.	0.11	S$:$CC$S	0.18

Table 1: TOP 5 MI Features by Writing Style

3.5 Syntax Tree Depth

Other research discovered the benefit of sentence length as feature either by computing the average sentence length [12, 13], or by using histograms over the sentence length [42]. Another simple but, to our knowledge, novel approach to distinguish different writing styles is to consider the depth of the syntax trees in the documents. We consider two approaches.

3.5.1 Statistical Moments

Statistical Moments are one way to characterize a distribution. The *k-th moment* of a random variable X is defined as $E(X^k)$. The expression $E([X-E(X)]^k)$ is called the *k-th central moment* of X.

For a given document d, containing n sentences (and so n trees), we can approximate the k-th moment and the k-th central moment as follows:

$$E(X^k) = \frac{1}{n}\sum_{j=1}^{n} x_j^k \tag{2}$$

and

$$E([X-E(X)]^k) = \frac{1}{n}\sum_{j=1}^{n} [X-E(X)]^k \tag{3}$$

where x_i is equal to the syntax tree depth of the i-th sentence of document d. Note, that $E(X)$ is known as the expectation value and $E([X-E(X)]^2)$ the variance of the random variable X.[1]

The values for different k vary in their order of magnitude. To avoid an overestimation of higher moments we introduce a normalization by taking the k-th root of the k-th moment. For the construction of the feature vectors we consider the first three moments and the second and third central moments [2]. Thus we can represent our document d as the following vector:

$$\Big(E(X), \sqrt{E(X^2)}, \sqrt[3]{E(X^3)}, \sqrt{E([X-E(X)]^2)}, \sqrt[3]{E([X-E(X)]^3)}\Big) \tag{4}$$

3.5.2 Histogram Approach

The most common form of a histogram is obtained by splitting the range of the data into equal-sized bins (called

[1]The fact that the central moment is not perfectly unbiased is not an issue for large n.
[2]The first central moment, $E([X-E(X)])$, is equal to 0.

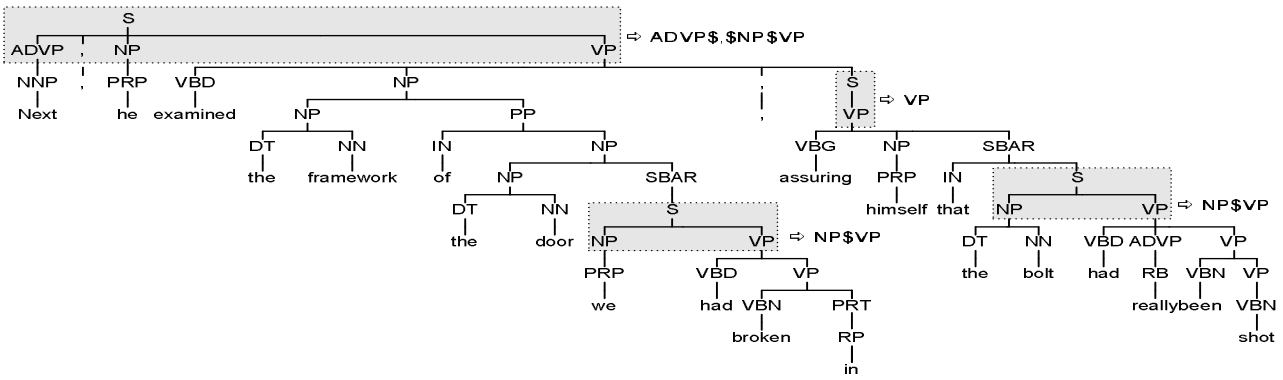

Figure 1: PCFG-Tree and Writing Style Features

classes). Then for each bin, the number of points from the data set that fall into the bin are counted [3].

In our scenario the data consists of the syntax tree depths of a document d. The value assigned to a bin is the number of trees within a certain range of depth (for example all trees of depth 10 to 12). Let $b(i)$ be the value of the i-th bin. As components of the feature vector for document d we consider these values normalized by the overall number n of trees in d and obtain the following vector:

$$\left(\frac{b(1)}{n}, \ldots, \frac{b(m)}{n}\right) \tag{5}$$

We used 5 as concrete bin-size in our implementation. Figure 2 shows a comparison between the tree depth distributions for the authors A. C. Doyle and R. Burton (again books from Gutenberg Project [1], see cpt. 5) in the form of histograms.

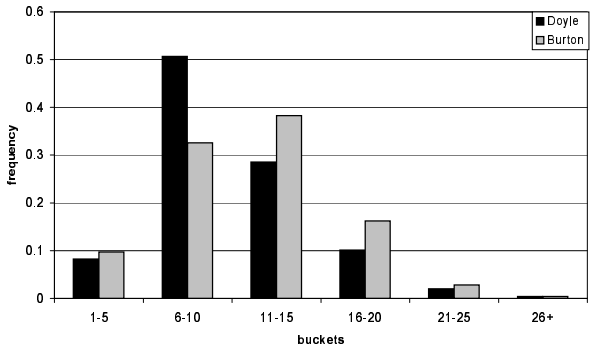

Figure 2: Tree Depth Histogram for two Authors

4. COMBINING FEATURES

In the previous section we have described several different document representations, providing us with different kinds of information about content and style. In this section we describe two approaches to put these pieces of information together.

4.1 Combination Vectors

The idea of combination vectors is to merge the vectors obtained by different document representations into a single vector. This can be done by the concatenation of feature spaces. More precisely we are given k vector representations

$$\vec{v_1}(d), \ldots, \vec{v_k}(d) \tag{6}$$

for document d with

$$\vec{v_i}(d) = (v_{i1}(d), \ldots, v_{im_i}(d)) \tag{7}$$

where m_i is the size of the feature space for the i-th representation.

These vectors can be combined into a combination vector as follows:

$$\left(\frac{v_{11}(d)}{n_1}, \ldots, \frac{v_{1m_1}(d)}{n_1}, \ldots\ldots, \frac{v_{k1}(d)}{n_k}, \ldots, \frac{v_{km_k}(d)}{n_k}\right) \tag{8}$$

Here the values n_i are normalization constants. The rationale for this normalization is that strong variations between the order of magnitude of the components of the feature vectors might occur (this holds, e.g., for Bag-of-Words vs. moments of syntax tree depth distributions). We choose the normalization constants such that the average component value is the same for all subspaces corresponding to the original feature spaces. Formally, for a document set D, we choose the constants n_i such that the following requirement is satisfied:

$$\frac{1}{n_i}\frac{1}{m_i}\sum_{d\in D}\sum_{l=1}^{m_i} v_{il}(d) = \frac{1}{n_j}\frac{1}{m_j}\sum_{d\in D}\sum_{l=1}^{m_j} v_{jl}(d)$$
$$\text{for all } i,j \in \{1,\ldots,k\} \tag{9}$$

We can assign one of the n_i an arbitrary value (say $n_1 = 1$); then the other normalization constants can be computed by elementary transformations of equations 9.

4.2 Meta Classification

For meta classification we are given a set $V = \{v_1, \ldots, v_k\}$ of k binary classifiers, obtained by supervised learning based on the features for distinct document representations, with Results $R(v_i, d)$ in $\{+1, -1, 0\}$ for a document d, namely, $+1$ if d is accepted for the given topic by v_i, -1 if d is rejected, and 0 if v_i abstains. We can combine these results into a meta result: $Meta(d) = Meta(R(v_1, d), \ldots, R(v_k, d))$

in $\{+1, -1, 0\}$ where 0 means abstention. A family of such meta methods is the linear classifier combination with thresholding [39]. Given thresholds t_1 and t_2, with $t_1 > t_2$, and weights $w(v_i)$ for the k underlying classifiers we compute $Meta(d)$ as follows:

$$Meta(d) = \begin{cases} +1 & \text{if } \sum_{i=1}^{n} R(v_i, d) \cdot w(v_i) > t_1 \\ -1 & \text{if } \sum_{i=1}^{n} R(v_i, d) \cdot w(v_i) < t_2 \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

This meta classifier family has some important special cases, depending on the choice of the weights and thresholds:
1) voting [8]: Meta returns the result of the majority of the classifiers.
2) unanimous decision: if all classifiers give us the same result (either +1 or -1), Meta returns this result, 0 otherwise.
3) weighted averaging [43]: Meta weighs the classifiers by using some predetermined quality estimator, e.g., a leave-one-out or k-fold-crossvalidation estimator for each v_i.

The restrictive and tunable behavior is achieved by the choice of the thresholds: we dismiss the documents where the linear result combination lies between t_1 and t_2. For real world data there is often a tradeoff between the fraction of dismissed documents (the *loss*) and the fraction of correctly classified documents (the *accuracy*). The idea of restrictive classification is to classify a subset of the test documents, but to do so with a higher reliability.

If a fixed set U of unlabeled documents (that does not change dynamically) is given, we can classify the documents with a user-acceptable loss of L as follows:

1. for all documents in U compute their classification confidence $\sum_{i=1}^{n} R(v_i, d) \cdot w(v_i)$
2. sort the documents into decreasing order according to their confidence values
3. classify the $(1-L)|U|$ documents with the highest confidence values according to their sign and dismiss the rest

In our experiments we assigned equal weights to each classifier, and instead of $R(v_i, d)$, we considered a "confidence" value $conf(v_i, d)$ for the classification of document d by the classifier. For SVM we considered the SVM scores, i.e., the distance of the test points from the hyperplane. A more enhanced method to map SVM outputs to probabilities is described, e.g., in [34].

5. EXPERIMENTS

5.1 Setup

For the validation of the presented techniques, we considered a literature data set obtained from the Gutenberg Project [1], a volunteer effort to digitize, archive, and distribute cultural works. We selected 10 English and American authors with a sufficient number of books (listed in Table 2). For each author we divided each book into parts with 20 paragraphs and stored each part as a document in the database. From these documents, we randomly choose 600 per class for our experiments. We divided these documents, that we obtained for each author a training set (100 documents) and an evaluation set (500 documents).

For our experiments we considered binary classification on all 45 possible pairs of authors (e.g. "Burton" vs. "Dickens"). For every pair we chose $T \in \{20, 40, 60, 80, 100\}$ documents from the authors' training sets as positive and the same number of documents as negative samples. The classification was performed on the union of both evaluation sets.

Then, we computed the micro-averaged *error*, i.e. the ratio of incorrectly classified documents to all test documents. For restrictive meta classification we considered in addition the *loss*, the fraction of documents dismissed by the restrictive classifier. Additionally, we computed the 95 percent confidence interval for the error.

We compared the following methods for feature construction:

1. word based features
   (a) Bag-of-Words using porter stemming and stopword elimination - see Section 3.1.1 (**BoW**)
   (b) Function words - see Section 3.1.2 (**FW**)
   (c) Part of Speech extraction of nouns and verbs; annotation with Connexor Machinese Phrase Tagger, using base forms of words constructed by Connexor - see Section 3.1.3 (**N&V**)
   (d) n-grams within constituents; using the Stanford Lexparser, considering constituents of each sentence represented as PCFG-tree - see Section 3.2 (**Constit.**)
2. structure based features
   (a) functional dependencies using Connexor Machinese Syntax for dependency tagging - see Section 3.3 (**FunctDep**)
   (b) writing style using the Stanford Lexparser - see Section 3.4 (**Style**)
   (c) histograms for syntax tree depth distribution - see Section 3.5.2 (**Hist.**)
3. combination vectors using Bag-of-Words, writing style, and tree depth histograms - see Section 4.1 - (**Combi**)

As classification method we chose standard linear SVM with parameter $C = 1000.0$. We used the popular *SVMlight* implementation [19].

5.2 Results

In our first experiment we compared the classification results of the different feature construction methods and their combination (see Table 3, Figure 3 for a chart representation of Bag-of-Words - the best base classifier - vs. combination methods). As meta method we used a simple unanimous decision (**Unanimous Decision**) classifier with base classifiers based on Bag-of-Words, writing style, and tree depth histograms.

In a second experiment we took the confidence values for classification into account and induced different loss values for the meta classification as described in Section 4.2 (Table 4 and Figure 4).

The main observations are:

- The stylistic features work significantly better than random; nevertheless Bag-Of-Words provides us with better results. Obviously, in the Gutenberg corpus there is a high correlation between authors and topics as well as distinct word pools.

Author	# Books	# Test Documents
Richard Burton	49	7425
Charles Dickens	55	7869
Arthur Conan Doyle	40	3473
Henry Rider Haggard	55	6882
George Alfred Henty	60	7169
Jack London	38	3566
Edgar Allan Poe	7	636
William Shakespeare	89	4025
Robert Louis Stevenson	45	6451
Mark Twain	129	9087

Table 2: Authors used from Gutenberg Corpus

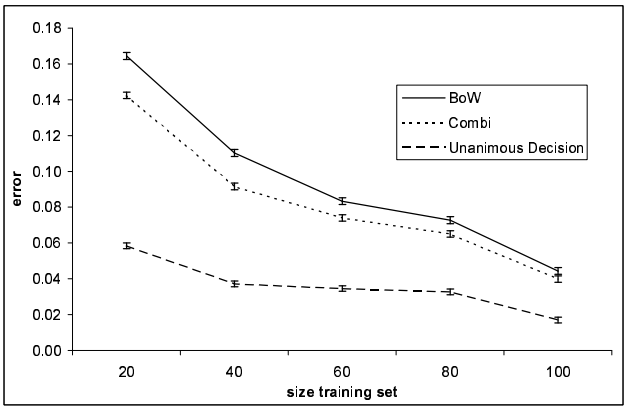

Figure 3: Comparison: Bag-of-Words and Combination Techniques on the Gutenberg Corpus

- By combining Bag-Of-Words with the alternative features, we obtained significant improvements. For combination vectors we have especially improvements for a low number of training documents. With restrictive meta methods we accept a certain loss, but obtain a much lower error on the remaining documents.

6. RELATED WORK

There is considerable prior work about alternatives to the Bag-Of-Words approach for document classification [28]. These include: using Part-Of-Speech (POS) tags ("verbs", "nouns", "adjectives", etc.) [33] either for disambiguation or for feature selection, using a thesaurus like Wordnet [15] for feature construction [38, 36], and feature selection based on statistical measures like Mutal Information or Information Gain [46]. N-grams of characters are popular for distinguishing different languages [10, 7]; also word based n-grams and phrases were examined for the text classification task [40, 24].

The problem of authorship attribution is different from the classical topic based classification task. Here, stylometric features may become important [17]. Baayen et. al. [4] show the occurrence of some kind of "stylistic fingerprint" for authors by considering a text corpus produced by student writers of different age and education level. They use the most frequent function words and apply principal component analysis (PCA) as well as linear discriminant analysis

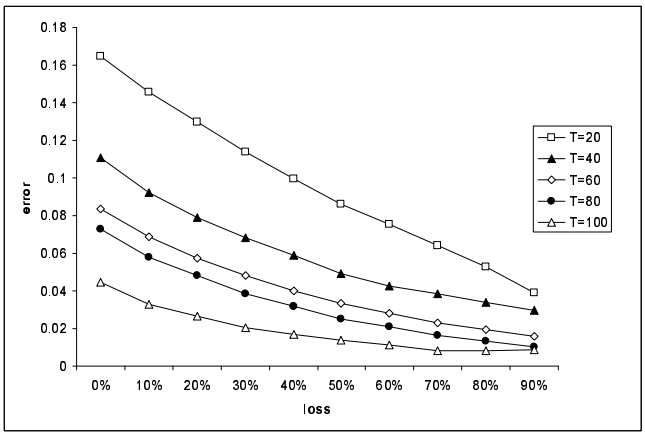

Figure 4: Comparison: Classification Results for Restrictive Meta Classification on the Gutenberg Corpus

Loss	T = 20	T = 40	T = 60	T = 80	T = 100
0 %	0.165	0.111	0.083	0.073	0.044
10 %	0.146	0.092	0.069	0.058	0.033
20 %	0.130	0.079	0.058	0.048	0.027
30 %	0.114	0.068	0.048	0.039	0.021
40 %	0.099	0.059	0.040	0.032	0.017
50 %	0.086	0.049	0.033	0.025	0.014
60 %	0.075	0.043	0.028	0.021	0.011
70 %	0.064	0.038	0.023	0.016	0.008
80 %	0.053	0.034	0.019	0.013	0.008
90 %	0.039	0.030	0.016	0.010	0.009

Table 4: Error for Different User-Provided Loss Values using a Meta Classifier with BoW, Writing Style, and Functional Dependencies on the Gutenberg Corpus

(LDA).

Diederich et al. [13] present a study on authorship attribution with Support Vector Machines. Their feature set consists of "full word forms" (in fact Bag-Of-Words) and so called tagwords, a combination of function words and grammatical information. Here, simple Bag-Of-Words outperforms their combination techniques with more enhanced linguistic features, in contrast to our combination vectors and meta methods.

In [5], Baayen et al. present a methodological study on the usefulness of stylometry-based features. They investigate features related to the writing style technique described above, taking grammatical rewriting rules derived from syntax trees into account.

The identification of unique users among a set of online pseudonyms using features such as simple words, misspellings, punctuation etc., is described in [31]. De Vel's work [12] deals with the exploration of style based features for identification of email authors. They use features such as style markers (average sentence or word length, total number of function words, vocabulary richness, etc.) and structural attributes (availability of signatures, number of attachments, etc.).

There are also several alternative learning paradigms for authorship attribution, e.g., Khmelev and Tweedie [21] con-

T	BoW error	FW error	N&V error	Style error	FunctDep error	Hist. error	Constit. error	Bigrams error
20	0.164 ±0.0034	0.354 ±0.0044	0.225 ±0.0039	0.186 ±0.0036	0.171 ±0.0035	0.299 ±0.0042	0.356 ±0.0044	0.458 ±0.0046
40	0.110 ±0.0029	0.240 ±0.0039	0.159 ±0.0034	0.138 ±0.0032	0.123 ±0.0030	0.278 ±0.0041	0.279 ±0.0041	0.390 ±0.0045
60	0.083 ±0.0026	0.177 ±0.0035	0.096 ±0.0027	0.123 ±0.0030	0.123 ±0.0030	0.273 ±0.0041	0.245 ±0.0040	0.323 ±0.0043
80	0.073 ±0.0024	0.147 ±0.0033	0.084 ±0.0026	0.114 ±0.0029	0.116 ±0.0030	0.275 ±0.0041	0.221 ±0.0038	0.285 ±0.0042
100	0.044 ±0.0019	0.115 ±0.0030	0.065 ±0.0023	0.103 ±0.0028	0.089 ±0.0026	0.272 ±0.0041	0.204 ±0.0037	0.230 ±0.0039

T	Combination error	Meta Unanimous Decision error	loss
20	0.142 ±0.0032	0.059 ±0.0016	0.482
40	0.092 ±0.0027	0.037 ±0.0014	0.399
60	0.074 ±0.0024	0.035 ±0.0013	0.362
80	0.065 ±0.0023	0.033 ±0.0013	0.349
100	0.040 ±0.0018	0.017 ±0.0010	0.345

Table 3: Error for Classification based on Different Features and their Combination on the Gutenberg Corpus

sidering learning models for authorship attribution tasks using Markov chains of characters, or Oakes [32] using a kind of swarm intelligence simulation technique called Ant Colony Optimization.

Combination vectors are used for authorship attribution (e.g. [42, 23, 13]), but neither explicit component weighting nor normalization are considered. The machine learning literature has studied a variety of meta methods such as bagging, stacking, or boosting [8, 44, 25, 16], and also combinations of heterogeneous learners (e.g., [47]). But, to our knowledge, meta classification was not applied in the context of authorship recognition.

7. CONCLUSION AND FUTURE WORK

In this paper we described classification with different document representations. In addition to well known features like document terms in the Bag-Of-Words model, POS tagging, etc., we considered alternative stylistic features like the depth or the structure of syntax trees. We combined the feature representations using two techniques: 1) combination vectors, where we constructed a single vector from the different feature vectors with automatically normalizing the combination vector's components so that the average component value is the same for all subspaces, 2) meta methods combining the classification results based on the different representations into a meta result. Our experiments on the author recognition task show that our new features are suitable for discriminating different styles and, used within combination techniques, lead to significant improvements of the classifier performance.

Our ongoing and future work includes a number of relatively obvious directions like 1) the improvement of existing and the construction of new alternative features, 2) application of different feature spaces and their combination for clustering, 3) the use of enhanced features for expert queries in specialized search engines.

8. REFERENCES

[1] Gutenberg project. *http://www.gutenberg.org/*.

[2] Lexparser. *http://www-nlp.stanford.edu/downloads/lex-parser.shtml*.

[3] National institute of standards and technology.

[4] H. Baayen, H. van Halteren, A. Neijt, and F. Tweedie. An experiment in authorship attribution. *JADT*, 2002.

[5] H. Baayen, H. van Halteren, and F. Tweedie. Outside the Cave of Shadows: Using Syntactic Annotation to Enhance Authorship Attribution. *Literary and Linguistic Computing*, 11(3):121–131, 1996.

[6] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[7] K. R. Beesley. Language identifier: A computer program for automatic natural-language identification on on-line text. In *29th Annual Conference of the American Translators Association*, 1988.

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

[9] C. Burges. A tutorial on Support Vector Machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 1998.

[10] W. B. Cavner and J. M. Trenkle. Text categorization and information retrieval using wordnet senses. In *Third Annual Symposium on Document Analysis and Information Retrieval*, 1994.

[11] S. Chakrabarti. *Mining the Web*. Morgan Kaufmann, 2003.

[12] O. de Vel, A. Anderson, M. Corney, and G. Mohay. Mining e-mail content for author identification forensics. *SIGMOD Rec.*, 30(4):55–64, 2001.

[13] J. Diederich, J. Kindermann, E. Leopold, and G. Paass. Authorship attribution with support vector machines. *Applied Intelligence*, 19(1-2):109–123.

[14] S. Dumais and H. Chen. Hierarchical classification of Web content. *SIGIR*, 2000.

[15] C. Fellbaum. *WordNet: An Electronic Lexical*

Database. MIT Press, 1998.

[16] Y. Freund. An adaptive version of the boost by majority algorithm. *Workshop on Computational Learning Theory*, 1999.

[17] D. Holmes. The Evolution of Stylometry in Humanities Scholarship. *Literary and Linguistic Computing*, 13(9):111–117, 1998.

[18] D. Holmes and R. Forsyth. The Federalist Revisited: New Directions in Authorship Attribution. *Literary and Linguistic Computing*, 10(2):111–127, 1995.

[19] T. Joachims. Text categorization with Support Vector Machines: Learning with many relevant features. *ECML*, 1998.

[20] T. Joachims. A statistical learning model of text classification for Support Vector Machines. *SIGIR*, 2001.

[21] D. Khmelev and F. Tweedie. Using Markov Chains for Identification of Writers. *Literary and Linguistic Computing*, 16(3):299–308, 2001.

[22] D. V. Khmelev and W. J. Teahan. A repetition based measure for verification of text collections and for text categorization. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 104–110, New York, NY, USA, 2003. ACM Press.

[23] M. Koppel, S. Argamon, and A. Shimoni. Automatically Categorizing Written Texts by Author Gender. *Literary and Linguistic Computing*, 17(4):401–412, 2002.

[24] D. Lewis. *Representation and learning in information retrieval*. PhD thesis, Department of Computer and Information Science, University of Massachusetts, 1992.

[25] N. Littlestone and M. Warmuth. The weighted majority algorithm. *FOCS*, 1989.

[26] C. Manning and H. Schuetze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[27] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1994.

[28] A. Moschitti and R. Basili. Complex linguistic features for text classification: A comprehensive study. In *ECIR*, 2004.

[29] F. Mosteller and D. Wallace. *Inference and Disputed Authorship: The Federalist*. Addison-Wesley, 1964.

[30] N. Nanas, V. Uren, and A. de Roeck. Learning with positive and unlabeled examples using weighted logistic regression. In *15th International Workshop on Database and Expert Systems Applications(DEXA'04), Zaragoza, Spain*, 2004.

[31] J. Novak, P. Raghavan, and A. Tomkins. Anti-aliasing on the web. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 30–39. ACM Press, 2004.

[32] M. Oakes. Ant colony optimisation for stylometry: The federalist papers. In *Proceedings of the 5th International Conference on Recent Advances in Soft Computing*, pages 86–91. Nottingham Trent University, 2004.

[33] B. Pang and L. Lee. Thumbs up? sentiment classification using machine learning techniques. In *EMNLP*, 2002.

[34] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers, MIT Press*, 1999.

[35] M. Porter. An algorithm for suffix stripping. *Automated Library and Information Systems*, 14(3).

[36] P. Rosso, E. Ferretti, D. Jimenez, and V. Vidal. Text categorization and information retrieval using wordnet senses. In *The Second Global Wordnet Conference GWC*, 2004.

[37] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management, 1988, p. 513-523.*

[38] S. Scott and S. Matwin. Text classification using wordnet hypernyms. In *Workshop on Usage of Wordnet in Natural Language Processing Systems*, 1998.

[39] S. Siersdorfer, S. Sizov, and G. Weikum. Goal-oriented methods and meta methods for document classification and their parameter tuning. In *ACM Conference on Information and Knowledge Management (CIKM 04), Washington*, 2004.

[40] C. M. Tan, Y. F. Wang, and C. D. Lee. The use of bigrams to enhance text categorization. *Information Processing and Management., vol. 30, No. 4, pp. 529-546, 2002.*

[41] P. Tapanainen and T. Jõrvinen. A non-projective dependency parser. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, 1997.

[42] H. van Halteren. Writing Style Recognition and Sentence Extraction. *Workshop on Text Summarization, DUC*, 2002.

[43] H. Wang, W. Fan, P. Yu, and J. Han. Mining concept-drifting data streams using ensemble classifiers. *SIGKDD*, 2003.

[44] D. Wolpert. Stacked generalization. *Neural Networks, Vol. 5, pp. 241-259*, 1992.

[45] Y. Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1(1/2), 1999.

[46] Y. Yang and O. Pedersen. A comparative study on feature selection in text categorization. *ICML*, 1997.

[47] H. Yu, K. Chang, and J. Han. Heterogeneous learner for Web page classification. *ICDM*, 2002.