Feature Engineering for Author Profiling and Identification: On the Relevance of Syntax and Discourse

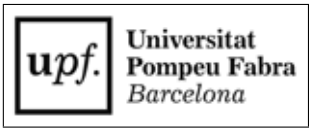
Juan Soler-Company

TESI DOCTORAL UPF / ANY 2017

DIRECTOR DE LA TESI

Dr. Leo Wanner

DEPARTAMENT DE TECNOLOGIES DE LA INFORMACIÓ I LES COMUNICACIONS



To myself, I deserve it.

Acknowledgements

I would like to thank my family for their unconditional support. They have always believed in me and are the main reason I have been able to achieve everything that I achieved.

My friends from Menorca, which are really looking forward to calling me Dr. Soler, need to be mentioned as well: Massi, Raul, Mateu, Pedro, Borja, Leire, Maria, Patri, Susana, Yudes, Deivid... every one of them is special and deserves a small space in this thesis.

I would like to thank the UPF in general, and the DTIC department for letting me teach these years. Teaching undergrad students has been an amazing experience, I learned a lot from the students and I hope they also learned a lot from me. Teaching is one of my new passions, and definitely one of the highlights of these four years.

Of course I need to thank Leo Wanner for his supervision throughout this years of madness. TALN members such as Luis, Francesco (both the good and the bad one), Michos, Simon, Ali, Rupero, Janine, Ahmed, Horacio... all need to be thanked, their wisdom and company in Kurz and Gut (sacred temple of drunkenness) have had great influence in this work.

Mónica deserves her own paragraph: she is the unexpected MVP of these 4 years, we started a great project together, we form a great team and my happiness has reached levels I didn't think existed. All because of you. I love you.

Finally, I would like to thank the members of the examining board, Paolo Rosso, Walter Daelemans and Horacio Rodríguez for accepting to serve on the committee.

Abstract

Author profiling and identification are two areas of data-driven computational linguistics that have gained a lot of relevance due to their potential applications in, e.g., forensic linguistic studies, marketing analysis, and historic/literary authorship verification. Author profiling aims to identify demographic traits of the authors, while author identification aims to identify the authors themselves by searching for distinctive linguistic patterns that distinguish them. The majority of approaches in the related work tends to focus on the content of the texts. We argue that focusing on structure rather than content can be more effective. The main focus of the thesis is thus on feature engineering, the development, evaluation and application of the feature set in the context of machine learning techniques to author profiling and identification. We prove the profiling potential of syntactic and discourse features, which achieve state-of-the-art performance in many different scenarios, especially when combined with other features.

Resum

El perfilament i la identificació d'autors són camps de la lingüística computacional que han guanyat rellevància als últims anys gràcies a les seves potencials aplicacions al camp de la lingüística forense o a la verificació d'autoria de textos històrics. El perfilament d'autors té com a objectiu identificar trets demogràfics dels autors; la identificació d'autors tracta d'identificar l'autor del text. Per fer-ho, es busquen automàticament patrons lingüístics per diferenciar entre autors/trets demogràfics. La majoria de treballs anteriors, es centren en el contingut dels texts. Nosaltres argumentem que analitzar l'estructura del text pot ser una alternativa més efectiva. El focus d'aquesta tesi està per tant, al *feature engineering*: la extracció, avaluació i utilització d'un conjunt de característiques lingüístiques amb algoritmes d'aprenentatge automàtic per a perfilar/identificar autors. Demostrem que les característiques sintàctiques i discursives són rellevants i que combinades amb altres, obtenen resultats a l'altura de l'estat de l'art.

Índex

Índex de figures

Índex de taules

Chapter 1

INTRODUCTION

In the last 20 years, Internet has evolved from a network of connected computers used to share data among researchers, to a significant part of everybody's lives. The growth of the net has been impressive: the number of Internet users grew from 70 millions in 1997 to 3,675 millions in 2016, which represents more than half of the world's population[1]. The number of indexed websites has grown from 1,117,255 in 1997 to 863,105,652 in 2015[2]. As a result of this growth and the birth of social networks, blogs and many other websites where users are given the opportunity of easily creating or uploading content, the amount of data that is generated every day has also grown immensely.

To put things into perspective, in 2015, around 2.5 quintillion bytes of data were created every day, which would fill 10 million Blu-ray disks, which, if stacked, would be four times taller than the Eiffel Tower[3]. Every minute, 216K Instagram posts are created, 204M emails are sent, 12h of video are uploaded to Youtube, and 277K tweets are posted. Most of the generated data in the net is thus unstructured.

One of the characteristics of the Internet nowadays is that a user can post anonymously in forums, comment sections of articles, social networks, chat systems, etc. By "anonymously", we mean that the person behind the computer does not need to reveal – in most cases – any personal information about who he/she is in real life (and if this type of information is required, it is easy to lie). This has positive and negative effects: on the one hand, users do not need to reveal who they are, obtaining, in theory, a way to express themselves with freedom; on the other hand, this freedom is often used to insult, threaten or troll fellow Internet

[1] Source: http://www.internetworldstats.com/emarketing.htm

[2] Source: http://www.internetlivestats.com/total-number-of-websites/

[3] Source: https://storageservers.wordpress.com/2016/02/06/how-much-data-is-created-daily/

users.

The effects of trolling users is much more severe than we might expect. Cyberbullying is a new sort of bullying that has emerged in online platforms. Examples of cyberbullying include mean/menacing text messages or emails, fake rumors spread by email/social media/messaging apps, and embarrassing pictures. The US National Center for Education Statistics and Bureau of Justice Statistics indicated that in 2015, about 21% of students between 12-18 years old had experienced cyberbullying in the U.S.A.[4]. It is clear that this is a huge issue that needs to be addressed. Even though this is not the main aim of this thesis, our work can help identify offenders and profile anonymous users to prevent them from harassing other users.

A question that we should ask ourselves now is: How can we prevent this sort of behaviour caused by the apparent anonymity that the net provides? Are people really anonymous when posting online? Isn't there anything in the message itself that could be analyzed to extract information about the author of the text? The answer of these questions is "yes"; several characteristics of a message can be extracted and used to classify the writer of the text with respect to different criteria. This is where author profiling and author identification come to play.

1.1 Author Profiling and Identification

Author profiling is based on the hypothesis that authors with similar demographic traits express themselves in terms of common linguistic patterns because they have been exposed to similar influences. These linguistic patterns can be extracted and used to classify the author of a text by demographic characteristics such as gender, age, native language or sexual orientation.

Author identification deals with the identification of the author of a text, given a predefined set of authors. The hypothesis behind this task is that the linguistic style of an author is unique enough to be distinguished from the styles of other authors.

To perform both tasks, a feature engineering process needs to be carried out. Both author profiling and identification are based on the determination and extraction of features that characterize the writings with respect to their author or with respect to the characteristics of the author. A carefully chosen feature set will be able to distinguish between authors/demographic traits, an ineffective feature set will not be able to do so.

Being able to identify the author of a text or, at least, characteristics of the author, is a very attractive idea, even more so because the areas of author profil-

[4]Source: `https://nces.ed.gov/pubs2017/2017015.pdf`

ing and author identification have many practical applications. Author profiling and identification can be applied to different fields such as forensic linguistics, literary and historic studies, and marketing. In the field of forensic linguistics, author profiling and identification could be applied to threatening letter analysis, to pedophile detection in chat systems, and to the automatic detection of cyberbullying. In the case of literary and historic studies, author identification can be applied to confirm/refute the authorship of a text by a specific author (the authorship of Shakespeare's plays and the authorship of the Federalist papers have been discussed in author identification literature for a long time). The automatic profiling of the authors of user feedback could be also very useful in marketing studies; detecting demographic patterns is very helpful in order to adapt the product/service that is being sold to the specific target audience.

1.2 Motivation of the Thesis

There have been many previous approaches that implemented author profiling and author identification systems successfully. These approaches have the tendency of focusing on the content of the text instead of on its inner structure. They can be very effective in controlled environments, where every document in the dataset belongs to the same genre and domain, is written in the same language, and has similar characteristics. However, content-related approaches do not generalize well and are computationally expensive (using feature vectors of thousands of features).

It is thus desirable to come up with an approach that uses less features and can still compete with state-of-the-art proposals in terms of performance. This goal can be achieved only if more distinctive and more generic features than those commonly used in the literature are exploited. Features of this kind are likely to be rather of a structural than content-oriented nature.

The main focus of our work is thus on feature engineering. We believe that a carefully chosen set of features that is able to characterize the writing style of the authors can be a very effective approach that circumvents some of the shortcomings that previous approaches had.

However, before going any further, the term "writing style" must be defined. Writing styles reflect the ways that writers think about themselves and about writing, as well as the pattern of strategies that writers consistently use to achieve their goals, (Lavelle, 1997). Style includes diction (choice of words), tone, syntax, discourse, punctuation, spelling, voice and many other characteristics.

For a general analysis of writing style in English prose, see e.g., (Leech and Short, 2007; Leech, 2007); for a general study on linguistics and literary style, see (Freeman, 1970), and (Holmes, 1985) for a study on variables that might be

used as stylistic "fingerprints". Some examples of the mentioned fingerprints are the following characteristics: word-length, sentence-length, distribution of parts of speech, usage of function words, vocabulary richness, etc.

It has been shown that the writing style of an author evolves with time (Can and Patton, 2004), so the writing characteristics of writers are not necessarily static. Consider (Biber and Finegan, 1989), for a study on the stylistic evolution of essays, letters and fiction writings over the last four centuries. In this study, the authors draw upon several relevant features of texts. The linguistic features that are considered include tense and aspect markers, pronouns and pro-verbs, questions, subordination/coordination features, etc. The authors conclude that a collection of features of this broad nature "can be used to compare other English varieties, for example, British and American writing, different styles of fiction, or, as in the present case, genres from different historical periods", which, as we will see, is similar to the reasoning behind our approach.

In addition to the mentioned features, Crystal and Davy (1969); DiMarco and Hirst (1993) state that syntactic features such as sentence structure and the frequency of specific phrasal or dependency patterns are relevant characteristics of the writing style of an author. Another group of features that are also of relevance are discourse structure features, according to (Burstein et al., 2003).

The majority of the above features are considered in this thesis.

A key goal of our system is to be versatile and easily applicable to different tasks. The feature set that is introduced in the following chapters has been successfully applied to different tasks such as gender, age, language, geographic origin and author identification. This makes us believe that it is a solid choice for many profiling tasks.

The approach that is presented is very effective in real-world applications due to the profiling potential of our feature set, which is able to perform well without needing huge amounts of data. A specific example of an application where our approach is useful is a system that automatically flags suspicious users in chat systems that are meant for under-aged children. In this situation, a potential pedophile could log in, change his/her word usage to lure their potential victims and remain unrecognized. If the system only checks word complexity and word usage, the offender might succeed; changing the words that are used is a relatively easy task. However, changing the inner structure of texts and many other stylistic markers is a much harder challenge. A user that constructs longer sentences with deeper syntactic trees than the majority of users in the system could be a person that is lying about his/her identity and that should not be let into the chat forum.

The characteristics of these real-world applications, where small training sets are available, make the presented feature set, combined with machine learning, an effective approach. Deep learning has been considered as an alternative classification method, but given the usual scarcity of data in practical applications, it

was discarded for the development of the thesis work (but will be explored in the future; see Chapter 6).

Before introducing the structure of the thesis, we summarize this introduction by explicitly stating the main goals of this thesis:

- To implement a system that effectively performs several author profiling and identification tasks.
- To do so by characterizing the writing style of the authors, analyzing not only word choices, but also deeper linguistic phenomena.
- To present a feature set that is small compared to most of the state-of-the-art approaches, but that obtains state-of-the-art performance.
- To test our feature set in different scenarios, varying genre, language and machine learning approach.
- To use feature selection methods in order to analyze the relevance of each feature in each presented experiment.

1.3 Structure of the Thesis

The thesis is structured into six chapters:

- Chapter 1 is the introduction to the thesis, where we present the task and motivate our approach.
- Chapter 2 explains the fundamental machine learning knowledge required to understand the experiments presented.
- Chapter 3 presents the state of the art in both author identification and profiling.
- Chapter 4 introduces, compares, evaluates, and details our feature set.
- Chapter 5 describes the experiments.
- Chapter 6 draws some conclusions from our work, points out some shortcomings of it, and outlines possible future work.

Chapter 2

FUNDAMENTALS

The goal of this chapter is to introduce the fundamental theoretical knowledge required to understand the experiments described afterwards in Chapter 5. All experiments that are presented in Chapter 5 use machine learning. Therefore, a basic understanding of machine learning techniques is required. In the following sections, we provide an introduction to machine learning in the context of author profiling/identification, to feature selection, and to evaluation and feature selection metrics.

2.1 Machine Learning in the Context of the Thesis

The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience. Machine learning focuses on the development of algorithms that learn from the provided data and extract underlying regular patterns from it. The extracted patterns can then be applied to the classification of unseen data instances.

Machine learning-based algorithms overcome some of the problems rule-based systems have, such as coping with unexpected input: in a rule-based system, an input that was not expected makes the system fail, and new rules need to be implemented to overcome this problem. A machine learning algorithm makes data-driven decisions automatically without the need of writing specific rules to solve the problem.

In the following subsections we define the different types of machine learning and present some of the most commonly used algorithms in the areas of author profiling and author identification.

2.1.1 Types of Machine Learning

Machine learning techniques are usually divided into three different groups: supervised, unsupervised and semi-supervised techniques.

Which of these three techniques is appropriate in each specific application depends on the amount of available correctly labeled instances: if there is plenty of correctly-labeled data, supervised learning is usually the chosen approach; if a small sample of labeled data is available, and a large amount of unlabeled data is easily obtainable, semi-supervised becomes the better option; if no labeled data is available, clustering or unsupervised learning is the only option.

2.1.1.1 Supervised Learning

Overview: Supervised learning is one of the most used techniques in machine learning. The goal of supervised learning is to build a model from a set of training data instances. In the model, each class is characterized by extracting common patterns from the feature vectors of the instances of that class, so if a new unseen instance is presented, the algorithm is able to make an informed prediction about the class this instance belongs to. Supervised machine learning approaches can have very competitive performance if sufficient amounts of labeled material are used. In some cases, the process of obtaining correctly labeled data is very costly: manual annotation is a time-consuming and expensive process and automatic annotation is not always an option.

The basic scheme of supervised machine learning is shown in Figure 2.1[1]. Let us discuss the individual steps of supervised machine learning following this scheme.

The figure shows that the problem definition stage, data gathering and preprocessing are the initial steps. These first steps are common to all machine learning techniques.

Once the problem is defined and the data is retrieved, the data needs to be split between a training set and a test set. The training set is the set of correctly labeled instances that are given to the classifier to train, i.e., to extract statistical patterns, in order to be able to predict the labels of unseen instances. The test set is used to assess to what extent the algorithm is able to predict the labels of any unseen instance. When the training set is chosen, it needs to be processed with a feature extractor, which extracts a set of characteristics from the texts. The chosen characteristics (or features) form a feature vector for each training set instance. Each vector dimension is a feature value. The training set feature vectors and their corresponding labels (or classes) are the input of the chosen supervised learning

[1] Image extracted from (Kotsiantis, 2007)

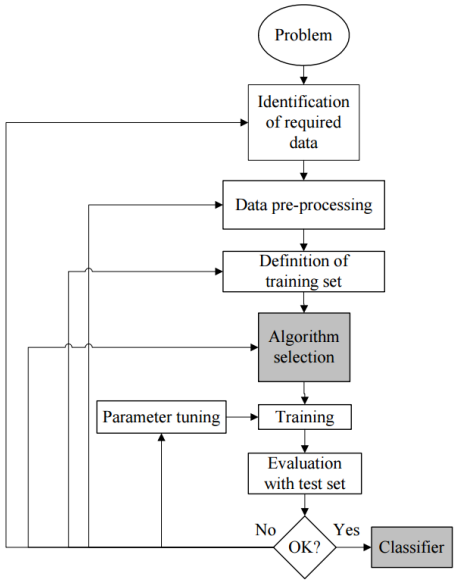


Figure 2.1: Supervised machine learning scheme.

algorithm. The classifier algorithm uses the data to create a model that characterizes the instances that belong to each one of the possible classes. As already pointed out, the model is later used to predict the classes of unseen instances. If the results are not optimal, two alternatives are often considered to improve the behavior of the algorithm: parameter tuning, where the parameters of the specific classifier are varied to ameliorate the performance of the system, or feature selection, where the performance of each feature and feature combination is analyzed and the best performing set of features is selected. Parameter tuning, although effective and in some cases necessary, has its problems: an overly tuned model might obtain the best performance for test instances that are similar to the training material, but the generalization power of the model might be low, as the model is too complex and is only able to fit the particular training data. If the issues are not fixed with the two mentioned methods, the feature set needs to be improved.

Each algorithm creates the model and extracts patterns from the input data in a different way. As a result, each algorithm has its advantages and disadvantages, and evaluating which classifier is the most suitable for the faced problem is necessary for achieving the best possible performance.

To illustrate how supervised machine learning works, in what follows, some of the most common algorithms in author profiling and identification are introduced. The algorithms that are introduced are: Naïve Bayes, K Nearest Neighbors, Sup-

port Vector Machines, Decision Tree and Random Forests.

Naïve Bayes (NB): NB classifiers are a family of algorithms that are based on applying Bayes' Theorem. NB classifiers assume that each one of the extracted features is independent from the other features.

The initial basic equation that is considered by this method is the following:

$$P(c|x) = \frac{P(x|c) * P(c)}{P(x)} \tag{2.1}$$

Since the denominator does not depend on c and the feature values are given, it can be considered a constant. The numerator is equivalent to the joint probability model. If we assume conditional independence between feature values, the final equation, which expresses the conditional distribution over the class variable c, is the following:

$$P(c_k|x_1, ..., x_n) = p(c_k) \prod_{i=1}^{n} p(x_i|c_k) \tag{2.2}$$

The classifier computes the probability of each possible class and assigns to a given instance the class with the highest probability. To better understand the usage of this classifier, an example is provided. Table 2.1 shows a training set, in which 500 texts written by men and 500 texts written by women are provided. From these texts, four different features are computed: the usage of commas, subordinate clauses, pronouns and passive voice. For each feature, we provide the number of instances which have high or low values per class. An instance has a high value of a feature if the feature value is higher than a predetermined threshold. If the feature value is lower than the threshold, we consider that the instance has a low value for that feature.

Training Set	**Commas**		**Subordinate**		**Pronouns**		**Passive**		**Total**
	High	Low	High	Low	High	Low	High	Low	
man	200	300	50	450	250	250	25	475	500
woman	400	100	400	100	350	150	450	50	500
Total	600	400	450	550	600	400	475	525	1000

Table 2.1: Simple training set example.

Given this training set and an unseen instance that has high values for each one of the features (i.e., each one of its feature values is higher than the corresponding feature threshold), let us apply the Naïve Bayes classifier to predict the class of this test instance. To do so, we compute the probability that this instance belongs to the class "man" respectively to the class "woman". The label with the highest

probability is the chosen one. The label "man" is referred to as "m" and the label "woman" as "w".

$$P(m|commaH, subordinateH, pronounH, passiveH) = P(commaH|m) * P(subordinateH|m) * P(pronounH|m) * P(passiveH|m) * P(m) \tag{2.3}$$

$$P(m|commaH, subordinateH, pronounH, passiveH) = \frac{200}{500} * \frac{50}{500} * \frac{250}{500} * \frac{25}{500} * \frac{500}{1000} = 0,0005 \tag{2.4}$$

As we can see, the probability that "man" is the chosen label, is quite low. Let us compute the corresponding probabilities for "woman".

$$P(w|commaH, subordinateH, pronounH, passiveH) = P(commaH|w) * P(subordinateH|w) * P(pronounH|w) * P(passiveH|w) * P(w) \tag{2.5}$$

$$P(w|commaH, subordinateH, pronounH, passiveH) = \frac{400}{500} * \frac{400}{500} * \frac{350}{500} * \frac{450}{500} * \frac{500}{1000} = 0,2016 \tag{2.6}$$

The probability that the test instance belongs to the category "woman" is much higher than the probability for the "man" label. As a result, the classifier predicts that the new instance is written by a woman.

Even though Naïve Bayes is a very simple algorithm, it usually performs competitively, even when compared with more sophisticated classification methods. One of the properties that makes this algorithm surprisingly useful is the decoupling of the class conditional feature distributions, which helps to alleviate problems caused by the curse of dimensionality (Keogh and Mueen, 2011). The main shortcomings of Naïve Bayes are that skewed datasets produce biased weights (in the sense that weights for classes with few training examples are smaller) and that the feature independence assumption, especially when using lexical content to classify, is too restrictive: each word contributes individually; therefore, their dependencies and correlations are not considered. Naïve Bayes classification has been used in many NLP tasks, including author profiling and identification, see e.g., (Altheneyan and Menai, 2014; Maharjan et al., 2014).

For further information about the algorithm and its surprising optimality, see e.g., (Rish, 2001; Zhang, 2004).

K Nearest Neighbors (kNN): kNN is another widely used algorithm. To understand this algorithm, the concept of *nearest neighbor* needs to be introduced. After the feature extraction process, each training instance is represented as a feature vector and a label. Given a test instance, its nearest neighbor is the training instance whose feature vector is the one that is spatially closest to the test instance. To compute the distance between instances, a distance metric needs to be chosen. Some examples of distance metrics are the Euclidean distance, the cosine distance, and the Manhattan distance.

The algorithm needs a positive integer as input, known as k, which indicates the number of nearest neighbors to consider.

To classify the unseen instances, the algorithm takes into account the classes of the k instances in the training set that are spatially closest to the test instance. Using a voting system, the classifier assigns the most common class among the nearest neighbors to the test instance.

A basic illustration of the process is shown in Figure 2.2[2], where the data is represented in a 2-dimensional space, and the instances belong to two different classes.

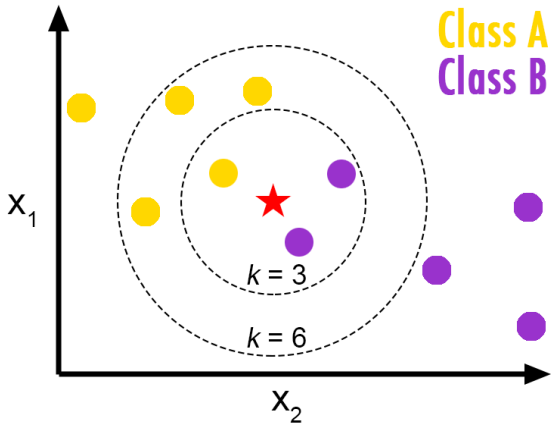


Figure 2.2: K nearest neighbors example.

The algorithm chooses class B, when k equals 3 and class A when k equals 6. It is obvious that the choice of k is crucial. A small value of k makes the algorithm more sensitive to noise. A large value makes it computationally expensive. A simple approach that is often used to choose the value of k is to select the square root of the number of instances.

Another important decision to be made when using this algorithm is the tie breaking criterion. Ties can occur if a simple voting system is used. A widely

[2]Source: `http://sciencepole.com/k-nearest-neighbors-algorithm/`

used criterion to break these ties is to weight the vote of the nearest neighbors with respect to the distance between the neighbor and the test instance. When this strategy is used, ties are much less likely.

The kNN algorithm is especially effective when small non-redundant feature sets are extracted. If high dimensional feature vectors are used, dimensionality reduction is usually performed to avoid the effects of the curse of dimensionality (Keogh and Mueen, 2011). The results of the algorithm are easily interpretable, and its predictive power is high, which is the reason why it is used in many different tasks, including author identification; see e.g., (Ghaeini, 2013).

Support Vector Machines (SVM): SVM is a supervised learning algorithm that builds a model which is used to classify the instances of two linearly distinguishable classes. To do so, the algorithm constructs a hyperplane that separates the instances of one class from the instances that belong to the other class by the largest margin possible. A simple diagram representing the basic SVM classifier is shown in Figure 2.3 [3].

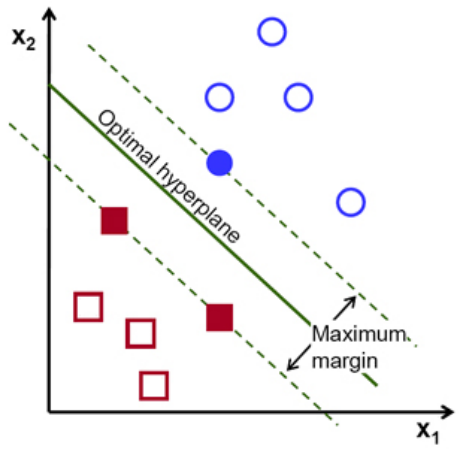


Figure 2.3: SVM simple classification diagram.

This is the simplest configuration of the algorithm. It is clear that this example is not realistic: in real applications, data is much more heterogeneous and, in the majority of cases, not linearly separable. To manage heterogeneous data, the algorithm is expanded to map the original finite-dimensional space into higher dimensional spaces to potentially make the separation easier. This technique is often called the "kernel trick". The kernel trick helps the algorithm perform non-linear classification efficiently. Each kernel (the function that maps the original

[3]Source: http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html

space into a higher dimensional space) is able to create boundaries of different shapes i.e., a linear kernel constructs a hyperplane that separates the classes, a radial kernel is able to construct hyperspheres, etc.

The algorithm is also used in scenarios where more than two classes are present. To adapt the algorithm to a multiclass scenario, there are two possible approaches: *one vs all* and *one vs one*. The first approach trains n binary classifiers that determine whether an example belongs to a particular class or not; the best performing classifier is chosen. The second one trains a binary classifier for each pair of classes, combining their outputs via a voting procedure.

The versatility of the SVM algorithm (many different kernels can be applied, depending on the type of data used) and its usually competitive performance makes it one of the preferred machine learning algorithms in author profiling and identification. The main drawbacks of SVM are that the choice of a kernel is critical (the performance of the system can change drastically depending on the chosen kernel), and that the results are hard to interpret, making SVM an algorithm often used as a "black box".

More information about the algorithm can be found in (Boser et al., 1992) and (Burges, 1998). Applications of the algorithm to author identification and author profiling are, e.g., (De Vel et al., 2001; Estival et al., 2007; Bayot and Goncalves, 2016).

Decision Tree (DT): DT learning techniques are also used in author profiling. The algorithm creates a tree from training data in which the intermediate nodes represent different feature value combinations, and the labels are assigned to the leaf nodes. An unseen instance is classified by traversing the tree depending on the test instance feature values. The reached leaf node determines the predicted label.

A simple example of a decision tree is shown in Figure 2.4. This is a real decision tree built using the J48 algorithm of the Weka machine learning toolkit (Hall et al., 2009). In this case, we are trying to learn whether a text is written by a man or by a woman. To do so, four simple features are considered: the mean number of commas per character, the mean number of characters per word, the question mark usage ratio, and the mean number of words per sentence. As we can see in the tree, the last feature is not considered: the decision tree algorithm did not find it relevant. We can also observe that depending on the other feature values, the tree assigns one label or the other to the test instance.

Let us now consider a test example and how the decision tree would classify it. Three different features are computed for the example: percentage of characters that are commas, the mean number of characters per word, and percentage of characters that are question marks. Let us assume that the feature values of

```
commas <= 0.004673
|   charsperword <= 5.201183: male
|   charsperword > 5.201183
|   |   questions <= 0.000057: male
|   |   questions > 0.000057: female
commas > 0.004673
|   charsperword <= 5.0199
|   |   commas <= 0.008471: male
|   |   commas > 0.008471: female
|   charsperword > 5.0199
|   |   questions <= 0
|   |   |   commas <= 0.006871: male
|   |   |   commas > 0.006871: female
|   |   questions > 0
|   |   |   charsperword <= 5.718182: female
|   |   |   charsperword > 5.718182
|   |   |   |   commas <= 0.020619: male
|   |   |   |   commas > 0.020619: female
```

Figure 2.4: Decision tree example.

the example are the following: commas:0.006, charsperword:6.2, questions:0.01. The instance would traverse the tree choosing the second "commas" branch, the second "charsperword", the second "questions" branch and inside that, the second "charsperword" and the first "commas" branch. As a result, the instance would be classified as a male writer.

The main advantages of this algorithm is that the decision tree created from the training set can be visualized, and, as a result, the outcomes are easy to understand and interpret. The DT algorithm requires little data preparation and is able to handle both numerical and categorical data. However, it also has several downsides. Thus, decision trees can create over-complex trees that overfit the training data and do not generalize very well. Biased trees can be built if the classes are very skewed, so it is advised to have a balanced number of instances per class. The algorithm can be sensitive to small variations in the data, which can change the resulting tree completely. To mitigate this sort of downsides, decision trees are usually used in ensemble approaches such as Random Forests (see immediately below).

Decision trees are implemented internally in different ways. An example of a widely used implementation of a decision tree is C4.5. For more information about C4.5, refer to (Quinlan, 2014). This algorithm has been used in several works on author identification, e.g., (Abbasi and Chen, 2005; Fissette, 2010).

Random Forests (RFs): RFs is an ensemble approach. Ensemble algorithms combine a group of weak classifiers to form a strong classifier. So, even if each of the weak learners by itself does not have the best predicting potential, the combination of their outputs does.

The algorithm uses a group of decision trees as weak learners. Each decision tree is formed using a random set of features. Given an unseen instance, its set of features serves as input for all the constructed decision trees. A voting system is used to classify the test instances: the class with more votes (each vote being the prediction of a decision tree) is the chosen prediction.

Figure 2.5[4] shows a diagram that illustrates the basic algorithm.

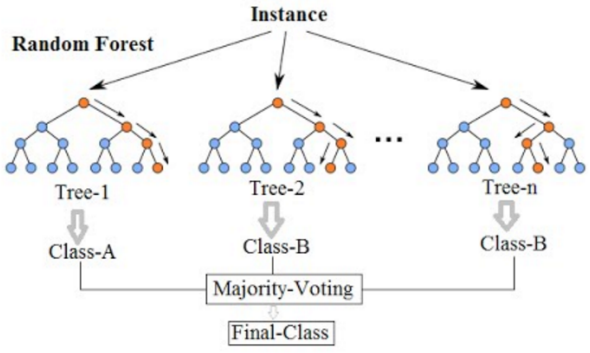


Figure 2.5: Random forests classification diagram.

As can be observed, each decision tree outputs a prediction, and the final decision takes into account the most common prediction.

The RFs algorithm needs some parameters to be tuned to achieve optimal performance. The number of trees that are built is one of them. More trees might give more predicting power to the system, but the computational cost also increases. Another parameter is the number of random features considered per tree. The square root of the number of features is often used as a rule of thumb, to determine the number of random features per tree. Finally, the type of the voting system is another important criterion to be established, especially for determining what happens when the system encounters a tie between two or more possible classes.

The algorithm is fast, achieves high accuracies and is used frequently in NLP and many other fields. The advantages that decision trees had are also advantages of this algorithm: categorical and numeric features are easily handled, little preprocessing of the features is required, and a very competitive algorithmic cost in

[4]Source: `http://www.ee.iitkgp.ac.in/ispschool/mvlss2016/`

the test phase, O(n log n), is achieved. At training time, even though the complexity of the algorithm depends on the number of constructed decision trees, the decision tree construction is a task that can be easily parallelized.

The problems that individual decision trees had, such as the tendency of overfitting the data, are solved by having many decision trees that consider random sets of features and combining the decision of each tree to make the final prediction.

For more information about the algorithm and an in-depth analysis of each possible configuration parameter, see (Breiman, 2001). Random forests have been applied to author profiling in (Palomino-Garibay et al., 2015) and to author identification in (Maitra et al., 2014).

2.1.1.2 Unsupervised Learning

A different machine learning paradigm that is widely used is unsupervised learning, also known as clustering. This paradigm is often applied to text classification applications and to author identification experiments, see, e.g., (Castillo et al., 2014; Leuzzi et al., 2013; Ferilli et al., 2015).

In both supervised and unsupervised learning, the feature extraction process is the same. As already mentioned above, the goal of the selected features is to characterize instances and to distinguish between different types of instances. The main difference between supervised and unsupervised approaches is that in the latter, the feature vectors that are formed after the feature extraction/selection process are not labeled, i.e., feature vectors do not have their corresponding categories assigned. Unsupervised learning compares feature vectors and groups them according to distance metrics between instances. Clustering algorithms partition the data in a way that the instances that are spatially close to each other form a group, or cluster of instances that have similar characteristics and that are different from instances that belong to other clusters. Each algorithm implements the notion of similarity and difference in a different way.

One of the most obvious advantages of this type of approach is that unlabeled data is much easier to gather than labeled data, which requires manual annotation in some cases. The main drawback of clustering is that its results are much harder to evaluate due to the lack of ground truth to compare to.

Unsupervised techniques can have very useful applications in author profiling/identification. Usually, the task of author identification is performed given a pre-defined set of authors. In a forensic linguistic application, this set might not be available and applying unsupervised techniques to find similarities between the investigated texts (anonymous threats, emails, kidnapping notes, etc.), and texts from closed-cases where the author was revealed, could be thus very the only option. The perpetrator will not be the same, but if the unknown instance is spatially close to, e.g., a group of under-aged males, the author of the text might have

similar demographic traits.

A specific application of unsupervised learning to the task of author identification that we implemented is shown in Figure 2.6.

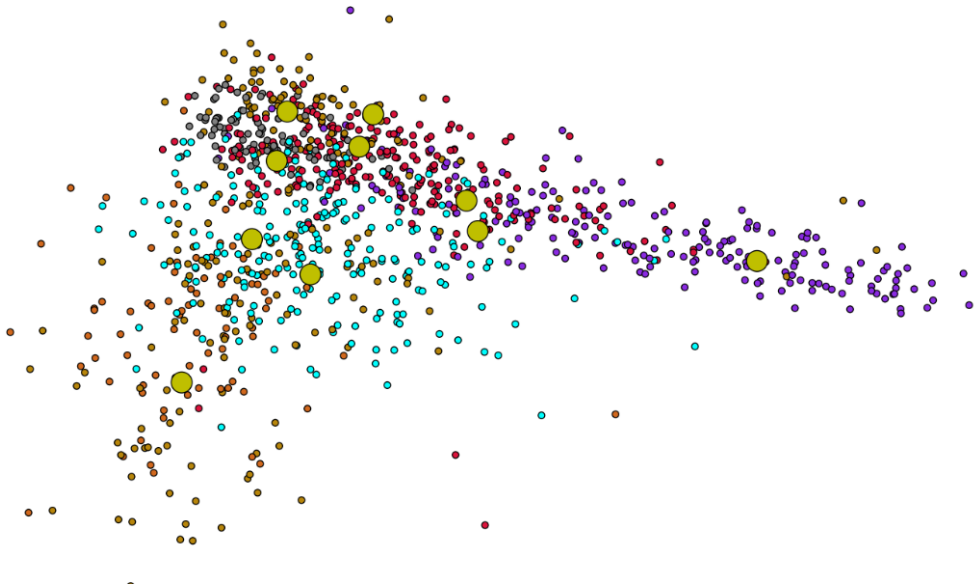


Figure 2.6: Author identification using clustering.

In the figure, each colored point represents a text. Each color corresponds to a different author. As far as features are concerned, the full feature set described in Chapter 4 is used. Each data point is the 2d representation of a feature vector formed by each of the features. The bigger yellow points are cluster centroids. A centroid can be defined as the center point of a group of instances. A simple way to compute the centroid of a cluster is to average the feature values of each feature vector, which results in a mean vector that represents the center of the cluster. As we can see in Figure 2.6, using our feature set, texts written by the same author are spatially closer to each other and distant from the writings of other authors. Therefore, using unsupervised techniques can be very useful to automatically form groups of instances that share either demographic traits or authorship.

The main choices that need to be made when using an unsupervised learning algorithm are a distance metric, a criterion to evaluate clustering configurations, and an algorithm that tries to optimize this criterion.

The proximity measures (or distances) that are often used are the same as mentioned before: Euclidean, cosine or Manhattan distance, among many others. The most optimal choice of the metric always depends on the kind of data that is being used. Therefore, this choice needs to be data-driven.

As was stated before, evaluating the performance of unsupervised learning algorithms is more challenging than in the supervised learning case due to the lack of ground truth to which the outcome can be compared. Two evaluation metrics that are often used are the compactness of the formed clusters and their isolation.

"Compactness" measures how close the data points are to the cluster centroid. A simple implementation of this metric is to compute the mean distance of each instance with its centroid (and this for each formed cluster). This value should be minimized by the algorithm, trying to group together the instances that are closest.

"Isolation" measures how separated the centroids of different clusters are. This value should be maximized; each cluster should be a separate group of instances that are close together and distant from other groups.

In most of the practical applications, expert judgment and specific evaluation techniques tailored to the specific problem are required to complement the introduced metrics.

After having briefly reviewed the characteristics of unsupervised learning, let us now introduce, the most popular clustering algorithm (K-Means) in the tasks of author profiling and identification.

K-Means (KM): KM is an algorithm that aims to group a set of multidimensional vectors into k clusters (with k being a parameter given a priori).

The basic flow of the algorithm is shown in Figure 2.7. k is set to three, such that three groups of instances must be formed by the algorithm. Each color indicates that an instance belongs to a cluster, so e.g., each black dot is an instance that forms the black cluster. The bigger red dots are the cluster centroids. The images represent the first 5 iterations of the algorithm (first five images), the iterations number 10, 15 and iteration 20 (which is the final iteration).

In the first step, k centroids are chosen. Several methods to choose the initial centroids are available. The most basic method is to choose k feature vectors randomly from the input data. After choosing the initial centroids, each vector is assigned to its nearest centroid, and the k centroids are recalculated as the average per dimension of each vector assigned to that particular centroid. This process is performed iteratively until no changes are made to the centroids.

One of the most critical parts of this algorithm is its initialization. A suboptimal initial choice of centroids (choosing, for example, k very close points) can lead to poor performance. To prevent poor initial choices, a good strategy is to choose k points that are far away from each other as initial centroids, and let the algorithm move them to find their optimal position.

Another important aspect of the algorithm is that outliers can highly impact the clustering process. To deal with outliers, one option is to detect them by es-

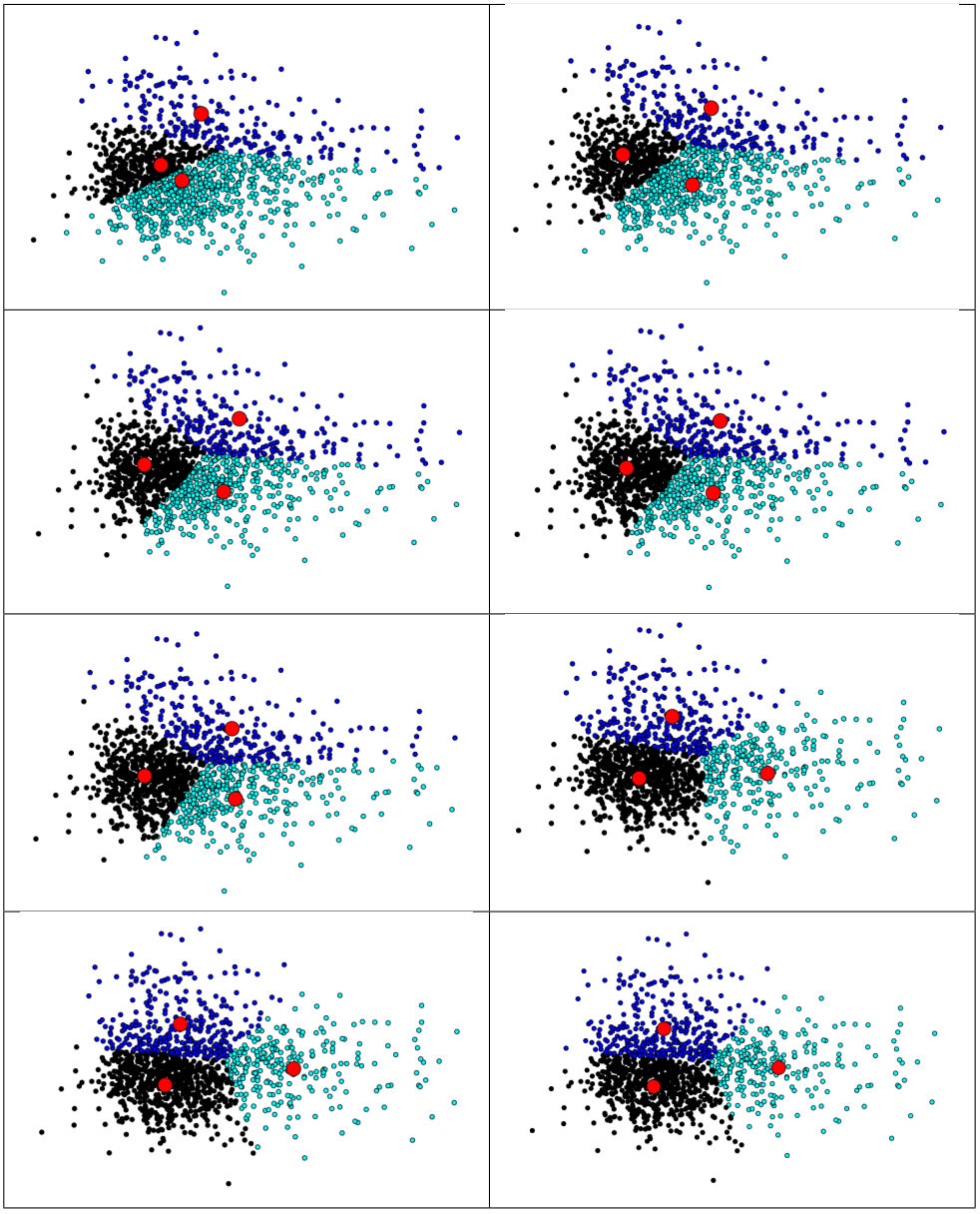


Figure 2.7: K-means example.

tablishing a distance threshold which, if exceeded, leads to the omission of the corresponding instance. Another option would be to recompute the centroids using the median instead of the mean. Using the median, the system would be less prone to be affected by outliers.

KM has been used in author profiling to cluster similar cyber-criminals by analyzing their created content, user preferences and online behaviour in (Zulfad-

hilah et al., 2016), to compute authorship similarity between email messages in (Chen et al., 2011), and to differentiate behavioral profiles of authors in (Meina et al., 2013).

For more information about the algorithm and its many possible variants, refer to (Jain, 2010).

2.1.1.3 Semi-Supervised Learning

Semi-Supervised learning is a machine learning technique that uses both labeled and unlabeled data to classify unseen instances. It is a less common type of machine learning, but can be very useful when labeled data is scarce and difficult to obtain and unlabeled data is easy to retrieve. The standard semi-supervised learning algorithm uses a small amount of labeled data and a large amount of unlabeled data to enrich the knowledge extracted from the labeled data. Two different phases are usually implemented: the enrichment phase and the test phase. During the enrichment step, labeled and unlabeled data are used. Different techniques are used to integrate the unlabeled instances. A possible approach called *self-training*, consists in classifying the unlabeled instances using the labeled instances to train, and integrating the most confident predictions to the labeled data, in order to enrich and expand the training set. It is very important to only introduce high confidence predictions to the training set, otherwise the introduced noise would worsen the performance of the system during the classification step instead of improving it. The classification phase uses the enriched corpus to predict the labels of a test set.

For a better understanding of the semi-supervised learning paradigm, an example is provided in Figure 2.8.

The blue circles represent texts written by men and the pink circles texts written by women. As we can see, the number of colored instances is quite low. We also have a larger amount of grey circles, which represent unlabeled instances, i.e., texts in which the gender of the author is unknown. The goal of the chosen semi-supervised algorithm is to enrich the dataset using the grey instances and then use the enriched data to classify unseen instances. The second image shows that some of the grey instances have been colored to reflect the most probable label. To perform this initial classification, we used a Nearest Neighbor classifier that considers the three nearest neighbors to decide. The colors are lighter, indicating that these instances have been added due to the enrichment process (or soft-labeled), and as a result, when classifying further instances, their vote should be weighted accordingly. One of the important decisions to be made when implementing this sort of algorithm is whether the enriched instances are used to classify, in further iterations, the remaining unlabeled instances. On the one hand, the soft-labeled instances can be helpful to further enrich the corpus if the initial predictions are accurate. On the other hand, adding to the training set noisy instances iteratively

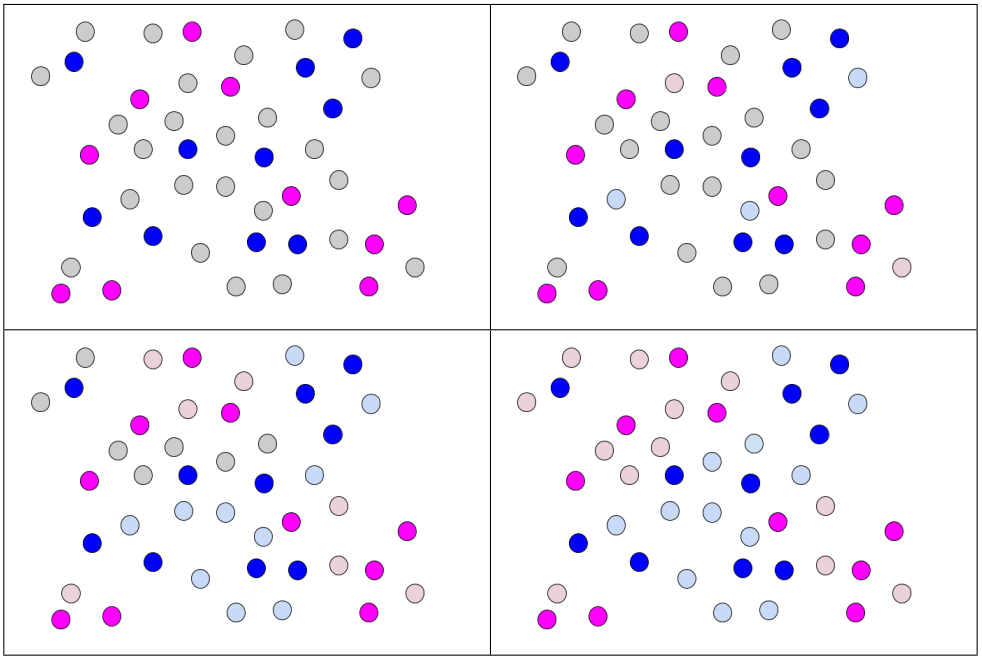


Figure 2.8: Semi-supervised learning enrichment phase example.

can be very detrimental. In the example, all unlabeled instances are added to the training set and used to enrich in further iterations. The third and fourth images show the end of the enrichment process. After this process, the training set has grown and a boost in performance is expected in the classification phase.

Semi-Supervised learning approaches have been used in author profiling and identification. See (Ikeda et al., 2008) for semi-supervised learning applied to gender and age identification using blog posts. Consider (Kourtis and Stamatatos, 2011) for an author identification approach in which unlabeled documents are added to the training set using a previous classification step.

2.2 Feature Engineering in Machine Learning

After introducing the different machine learning paradigms and algorithms that are often used in author profiling and identification, the topic of feature engineering needs to be discussed.

Feature engineering is a vaguely defined set of tasks related to designing feature sets for machine learning applications (which in some cases, is considered an art). The first important task to do, in order to correctly design a feature set is to understand the properties of the problem at hand and assess how they might interact with the chosen classifier. After understanding the problem, hypotheses

need to be drawn. Feature engineering is thus a cycle, in which a set of features is proposed, experiments with this feature set are performed, and, after analyzing the results, the feature set is modified to improve the performance until the results are satisfactory.

Although it is often possible to obtain competitive performance using fairly simple and obvious sets of features, there is room for significant performance improvement. Carefully constructed feature sets require thorough understanding of the task at hand, but can significantly outperform basic feature sets. In short, better features mean better results.

The data needs to be characterized by a group of features that differentiate between the instances that belong to a class with respect to the other classes. Irrelevant or partially relevant features can negatively impact the performance of the classifier. An example of an irrelevant feature would be one that takes a fixed value for any instance in the input data.

Optimal feature selection helps the algorithms extract patterns that generalize to unseen instances without needing complex parametrization of the classifier to perform competitively, preventing overfitting. Models created by the machine learning algorithm which contain the "knowledge" extracted from training data are faster to run, easier to understand and to maintain if the feature set is appropriate.

Different types of features can be extracted from the input data. Numeric features quantify numeric characteristics of the data; boolean features indicate presence or absence of a characteristic; and nominal features can take a fixed set of values (e.g., "positive", "negative", "neutral"). It is important to choose machine learning algorithms compatible with the type of features that are considered. In our specific case, the constructed feature set will be composed solely of numeric features. The specific feature set and its motivation, justification and evaluation are introduced in Chapter 4.

When numeric features are used, normalization has to be considered. In some cases, the feature values that compose a feature vector can have varying ranges, which can be detrimental to specific machine learning algorithms. An example of the influence of varying feature ranges is a simple clustering algorithm in which the Euclidean distance is used. The equation that is used to compute this distance is the following:

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + ... + (p_n - q_n)^2} \tag{2.7}$$

Imagine that instance q has the following feature vector: [0.01, 0.03, 29]. Instance p is represented by the following vector: [0.03, 0.11, 35]. The Euclidean distance between p and q is computed as follows:

$$d(p,q) = \sqrt{(0,03 - 0,01)^2 + (0,11 - 0,03)^2 + (35 - 29)^2} \tag{2.8}$$

$$d(p,q) = \sqrt{(0,02)^2 + (0,08)^2 + (6)^2} \tag{2.9}$$

$$\begin{aligned} d(p,q) &= \sqrt{0,0004 + 0,0064 + 36} \\ &= \sqrt{36,0068} \\ &= 6,0006 \end{aligned} \tag{2.10}$$

As we can see in the example, the distance metric is heavily influenced by the third feature, which has a larger numeric domain than the first two. To solve this problem, a normalization technique, which scales the value of the features to have a mean value of zero and a standard deviation of one, is *standardization*. Some algorithms are not sensitive to this data variability, so this process is not always mandatory; it depends entirely on the choice of the machine learning algorithm.

After this broad introduction, let us illustrate the feature engineering process with some examples.

First of all, let us think about the problem of gender identification. The goal is to distinguish between male and female authors. To create an effective feature set, we need to dig deep into the differences between genders. A possible hypothesis that could be formulated is that women tend to be more sensitive than men. To materialize this hypothesis in feature form, we can measure the sentiment of texts to prove it. To do so, a simple approach would be to use sentiment analysis lexicons to measure the percentage of words in a text that have positive or negative sentiment.

If we continue developing the previous hypothesis, we could also try to prove that women tend to write about how they feel about a certain situation, whereas men tend to write about the situation itself. A naïve approach to model this hypothesis would be to measure the usage of adjectives vs the usage of verbs: high values of the first measure could be caused by expressive writing, focused on giving an opinion on a topic, while high values of the second metric could be a result of a more action-based narrative. To see whether these features are distinctive, their values should be analyzed and their distinctiveness should be tested by either using feature relevance metrics such as Information Gain, or a machine learning algorithm to test the accuracy of the system with these features. A more in-depth example about author identification is introduced in Chapter 4.

Let us now give a further example. In this case, we are going to focus on the problem of automatic irony detection. The goal of this problem is to determine whether a text (in this case, a Tweet) contains ironic content or not. If we think about the characteristics of irony, we could say that unexpectedness can be considered a signal of irony. To measure unexpectedness, we could analyze the frequencies of the chosen words and see whether there is an imbalance: the use

of common words followed by rarely-used terms could signal ironic content. An example of this kind of imbalance is shown in the following Tweet:

"I believe Trump will do a fantabulous job."

It is clear that this Tweet is ironic due to the usage of "fantabulous", a word that is much less common than the surrounding ones. To measure imbalance, resources such as the British National Corpus could be used (Clear, 1993).

Another characteristic of the Tweets that can be used to determine whether the content is ironic or not is the specific choice of terms: choosing a term instead of one of its more common synonyms could indicate ironic intent. For example, if it is about to rain, someone might say "sublime weather today", choosing *sublime* over more common synonyms such as *nice*, *good* or *very good*. The listener might grasp this hidden information when he/she asks him/herself why a rare word like *sublime* was used in that context. Using the British National Corpus and WordNet (Miller, 1995), we could translate this idea into feature form. To see more examples of potential useful features for the irony detection example, see, e.g., (Barbieri and Saggion, 2014c,a,b).

2.3 Metrics

After introducing the different machine learning paradigms, the most frequently used algorithms and feature selection criteria, we need to introduce the evaluation and feature relevance metrics that will be used in the experiments carried out in the context of this thesis.

The evaluation metrics are used to evaluate the performance of our experiments. The feature relevance metrics show the relevance of a specific feature in a classification problem.

2.3.1 Evaluation Metrics

As was introduced before, in a supervised learning setting, the algorithm learns from a training set of correctly labeled examples. The extracted knowledge, also known as "model", is then applied to unseen instances. To evaluate the quality of a model, a set of unseen instances are classified, and the predictions of the classifier are compared with their real labels. Each evaluation metric implements this comparison in a different way. Some of the common metrics used to evaluate supervised machine learning models are the following: accuracy, precision, recall, and f-measure.

Accuracy is the most intuitive performance measure; it is defined as the ratio

of correctly predicted instances:

$$Accuracy = \frac{\#correct}{\#predictions} \tag{2.11}$$

This metric is mostly used in cases where the number of instances per class is evenly distributed.

Precision, recall, and f-measure are better performance metrics when class distribution is skewed. They take into account true/false positives/negatives. True positives/negatives ("tp", "tn") are defined as correctly predicted cases (if we consider two classes, one being positive and one being negative, a true positive is when a test instance is positive and the system predicts it). A false negative ("fn") is a case where the instance was positive and the system predicted incorrectly, and a false positive ("fp") is when the class was negative and the system predicted it to be positive. With these definitions, we can define:

$$Precision = \frac{tp}{tp + fp} \tag{2.12}$$

$$Recall = \frac{tp}{tp + fn} \tag{2.13}$$

Precision is thus a measure of how good the predictions with regard to false positives are, and recall measures how good the predictions are with respect to false negatives.

F-measure is the weighted average of precision and recall, expressed as follows[5]:

$$\textit{F-Measure} = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{2.14}$$

2.3.2 Feature Relevance Metrics

Feature relevance metrics are also very useful in the context of this work. These metrics help to understand how impactful each feature is in a supervised machine learning problem. A specific metric that has been often used in the context of the work of this thesis is *Information Gain*. Information Gain evaluates the relevance of an attribute or feature with respect to the classes that are being differentiated. To do so, Information Gain considers that an attribute is relevant when its value is in a specific range, the instances are classified as one class, and for another range, the instances are classified as a different class. An attribute is considered irrelevant if it provides no discrimination between classes.

[5]The equation computes the f-measure weighting precision and recall equally.

To introduce how Information Gain is calculated, we first need to introduce the concept of *entropy*. Entropy characterizes the purity of an arbitrary collection of examples. Given a collection S of positive and negative examples (2 different classes), the entropy of S in this binary classification scenario is defined as follows:

$$Entropy(S) = -p_+ \log_2 p_+ - p_- \log_2 p_- \tag{2.15}$$

where p_+ is the proportion of examples that belong to the positive class and p_- the proportion of examples that belong to the negative class. If n classes are considered, the entropy coefficient is computed as follows:

$$Entropy(S) = \sum_{i=1}^{n} -p_i \log_2 p_i \tag{2.16}$$

Information Gain is defined in terms of entropy. The Information Gain coefficient of a feature is the expected reduction in entropy caused by partitioning the examples according to this attribute. It is computed as follows:

$$Information\_Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{2.17}$$

where $Values(A)$ is the set of all possible values for attribute A, and S_v is the subset of S for which attribute A has the value v.

To compute the Information Gain of each feature, in our experiments we use the Weka toolkit (Hall et al., 2009). Figure 2.9 shows an example of the output that Weka provides.

The example shows the Information Gain of every feature in a classification problem, where British literary authors were classified according to their gender. The figure shows all considered features, sorted by their Information Gain. We can see that in this case the most distinctive features are the usage of curse words, verbs, annoyance-related words, and the ratio of modal verbs. Each feature that has an Information Gain value higher than zero contributes to the classification. A set of features is considered appropriate for a specific problem if the majority of features have Information Gain values higher than zero.

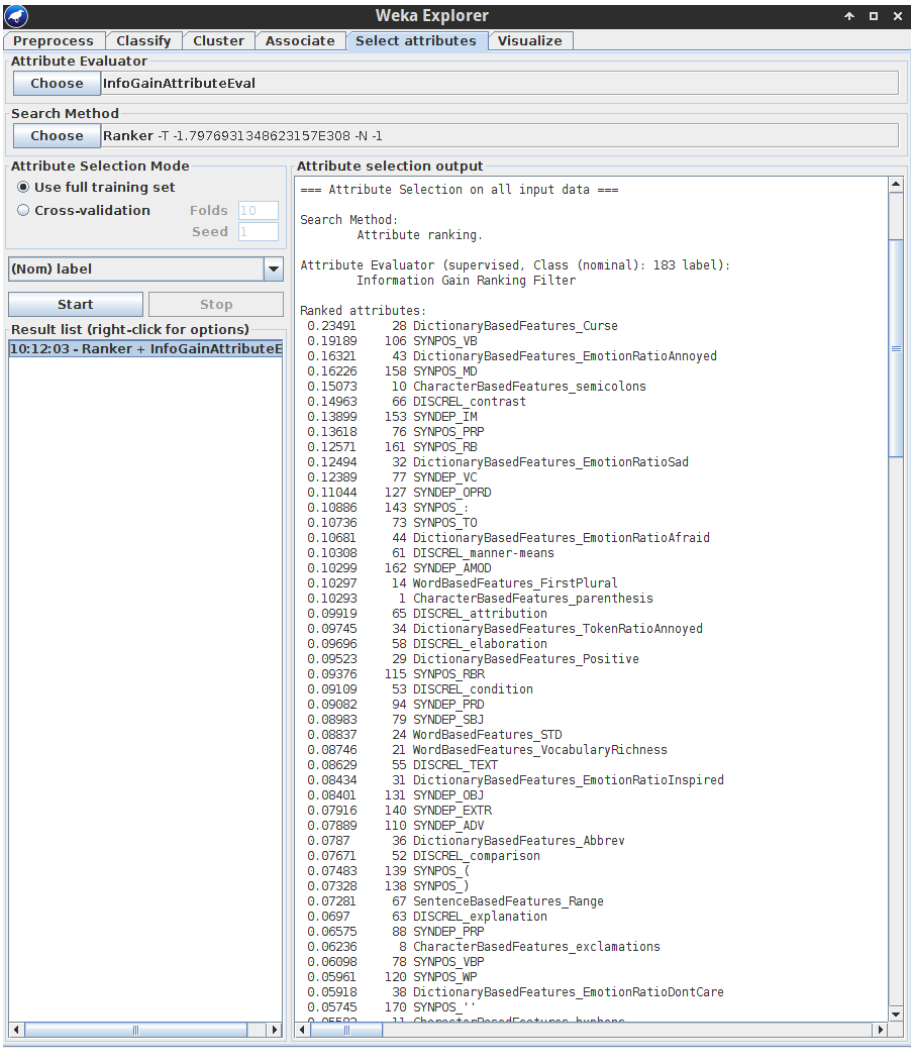


Figure 2.9: Weka's Information Gain of our feature set in a gender identification experiment.

Chapter 3

STATE OF THE ART

This chapter presents and discusses the approaches from the literature that are related to this thesis. Although author profiling is the main focus of the thesis, author identification is also considered, such that both author profiling and author identification approaches of the state of the art are discussed.

3.1 Author Profiling

As stated before, the goal of author profiling is to classify authors of written material with respect to their demographic characteristics.

There have been many attempts to perform author profiling using different approaches and different data types. The kind of data that is used is one of the main characteristics of each specific work: the features that are extracted are aimed to exploit some underlying tendencies of the chosen data and have to cope with the potential noise in the data (malformed sentences, orthographic errors, etc.). They thus highly influence the approach. Given this high relevance of data, we first review the datasets that have been used (as well as the most often used corpora and publicly available resources) and then focus on the approaches.

3.1.1 Datasets

Several different text types have been explored in the area of author profiling. We present each text type, as well as the relevant datasets.

3.1.1.1 Blog Corpora

One of the most explored datasets in author profiling are informal blog posts. The main reason behind this focus on blogs, is that there are huge amounts of data

created every day by bloggers, who write about many different topics. These bloggers, in many cases, reveal personal information about themselves. As a result, crawling blog posts and labeling them with the demographic information revealed by bloggers is not a challenging task.

"The Blog Authorship Corpus" is an example of a collection of informal blog posts that is often used to perform author profiling tasks.

The Blog Authorship Corpus

The Blog Authorship Corpus[1] is one of the resources that are publicly available and that have been used extensively. The corpus is introduced in (Schler et al., 2006). It is a collection of posts by 19,320 bloggers downloaded from the blog platform "Blogger". The corpus contains 681,288 posts, with over 140 million words. The blog posts are labeled with the gender, age, occupation and zodiac sign of the authors. It is an valuable resource because of its size, but some of its negative characteristics, such as noisiness, must be taken into account: the bloggers tend to paste chat conversations, music lyrics, or to create entries where only links are listed. There is also a considerable amount of spam text and the age labels can be unreliable (for instance, in one case, each entry of a 13-year-old is composed of ads and spam).

An analysis of the composition and characteristics of The Blog Authorship Corpus is provided in (Argamon et al., 2007). The 1,000 most frequent content words in the corpus are considered and used to perform an automated factor analysis of the usage rate of each one of these 1,000 words. The goal is to group them and create clusters of words that depict clear and distinct themes. Twenty clusters are formed: "conversation, "atHome", "family", "time", "work", "pastActions", "games", "internet", "location", "fun", "food/clothes", "poetic", "books/movies", "religion", "romance", "swearing", "politics", "music", "school", and "business". The mean frequencies per word group with respect to gender and age of the author are analyzed. One of the conclusions that is drawn is that the usage rate of words associated with "family", "religion", "politics", "business" and "internet" increases with age, while the usage of words associated with "conversation", "atHome", "fun", "romance", "music", "school", and "swearing" decreases significantly with age.

[1]Which is publicly available and can be downloaded from `http://u.cs.biu.ac.il/~koppel/BlogCorpus.htm`

Other Blog Corpora

Other resources of this type have been compiled and used in the field. Some examples are listed below.

Mukherjee and Liu (2010) compile a dataset composed of 3,100 texts from several popular blog sites (such as blogger.com and technorati.com) labeled by the gender of their author, which was determined by visiting the users' public profiles. 1,588 posts are labeled as written by men and 1,512 as written by women. The average post length is 250 words for men and 330 for women.

Sarawgi et al. (2011) compile a set of blog posts on seven specific topics: "education", "travel", "spirituality", "entertainment", "book reviews", "history", and "politics". For each topic, 20 articles written by authors of each gender are downloaded. From each blog, approximately the first 450 words, preserving the sentence boundaries, are kept. This resource avoids unwanted gender bias in topics by matching articles written by different genders even at the sub-topic level (if a blog post written by a man speaks about a specific TV show, the authors search for a female-written post about the same TV show).

Rosenthal and McKeown (2011) compile a dataset composed of 24,500 LiveJournal[2] blogs. A blog is only selected if the author provides his or her age publicly in the blog, is written by only one author, and the author is living in the U.S.
.

In (Pham et al., 2009), the authors present a corpus of 3,524 blogs written in Vietnamese, labeled with the gender, age ($\leq$ 22 , 23-26, $\geq$ 27), location (differentiating between North, South, and other) and occupation (e.g., "student", "singer", "model", etc.) of the bloggers. The collected texts must comply with the following criteria: 1) the author of a selected text must be a Vietnamese native speaker, 2) each chosen author must have more than ten entries, 3) only original content is retrieved, and 4) only blog entries written in the last 4 years are considered.

Oberlander and Nowson (2006) gather a corpus of personal blogs, where bloggers are required to answer socio-biographic and personality questionnaires. Participants are requested to submit one month's worth of prior weblog postings. The specific month is pre-specified so as to reduce the effects of a personal choice of the month. The corpus is composed of the writings of 71 participants (47 females and 24 males, averaging 27.8 and 29.4 years of age respectively).

"The Fisher Corpus" (David et al., 2004), is used in (Schler et al., 2006). This corpus contains 16,000 transcripts of telephone conversations (more than 2,000 hours of audio), labeled by the gender and age of the speakers. 38% of the subjects are of age 16-29, while 45% are of age 30-49, and 17% over 50.

[2] `http://www.livejournal.com/`

3.1.1.2 Email Data

Email data has also been used in author profiling. "The Enron Corpus" is the reference corpus of this type and the only one that is publicly available.

The Enron Corpus

The Enron Corpus[3] contains 517,431 emails from 150 authors (mostly senior managers of the Enron Corporation) that were published after an investigation of the Federal Energy Regulatory Commission, which led to the bankruptcy of the Enron Corporation, an American energy company based in Houston, Texas. Some of the works on this corpus are presented in Section 3.1.2.1.

Other Email Data

Estival et al. (2007) have collected a series of emails and label them by five demographic (age, gender, native language, level of education and country of residence) and five psychometric (agreeableness, conscientiousness, extraversion, neuroticism and openness) traits. The data has been collected using crowdsourcing. The participants were asked to donate ten email messages and to respond to a questionnaire that provides the labels to their texts.

3.1.1.3 Generic Reference Corpora

General discourse texts have also been used in author profiling. In (Koppel et al., 2003), the authors use "The British National Corpus" (henceforth, BNC) (Clear, 1993). The BNC is a 100 million word collection of samples of written and spoken language from a wide range of sources designed to represent a wide cross-section of British English, both spoken and written, from the late twentieth century. The written section of the BNC includes extracts from regional and national newspapers, journals, academic books, popular fiction, published and unpublished letters and essays, among other kinds of texts.

3.1.1.4 Social Media

The huge amount of data available in social networks and the common availability of personal information about the users (such as gender and age) makes the user-generated content in these platforms very viable for use in author profiling research. Twitter, Facebook and Netlog have been some of the social networks that were considered.

[3] Available at `https://www.cs.cmu.edu/~./enron/`

Twitter Data

Burger et al. (2011) use Twitter data. Their corpus contains approximately 213 million tweets from 18.5 million users. The data is multilingual, but with a predominance of English tweets. The tweets are automatically annotated by the gender of their author. To automatically annotate the tweets, only Twitter users with a blog associated were selected. The labels were extracted from the blog profile of the user. The final filtered version of the corpus contains more than 4 million tweets. This corpus is divided into a training set of 3.2 million tweets, a development set of 400K tweets, and a test set of 418K instances.

In (Nguyen et al., 2013), a collection of tweets written in Dutch is compiled. These tweets are manually annotated with the gender and age of the authors. The goal of the authors is to select a set of users as randomly as possible without biasing user selection by searching well-known stereotypical behaviour. Only accounts with less than 5,000 followers are considered to limit the inclusion of celebrities and organizations and only accounts with more than 10 tweets are selected.

These resources are not publicly available. A source of publicly available Twitter data is PAN, the shared task on author profiling, which every year releases a corpus of tweets (or content from other social networks), labeled by the gender and age of the authors. More information about the shared task is provided in Section 3.1.3.

Facebook Data

Facebook data is also used, although not as frequently as Twitter data. This may be due to the conditions of use of the Facebook API, which are much more restrictive than of the Twitter API.

Rangel and Rosso (2013a) present a dataset composed of Facebook comments written in Spanish. This choice is motivated by the freedom of expression and style that Facebook comments provide. Facebook data has also the advantage that the demographic information of a user is directly accessible and more reliable than in Twitter. Three different topics are considered: politics, football and celebrities. Comments without textual content are removed (i.e., comments that only contain emojis and links). Three annotators labeled these comments with the six basic emotions of Ekman's theory (Goldberg, 1993).

A dataset compiled along the same lines is presented in (Rangel et al., 2014a), where Facebook comments are labeled with the gender of the author (automatically), the emotion expressed in the comment (following the same six basic emotion labeling scheme), and the usage of irony in each comment (the last two labels

are assigned manually). The same topics as for the previous dataset are considered: football, politics and celebrities.

Netlog Data

Content generated by the Belgian online social networking platform Netlog[4] has also been used in author profiling. In (Peersman et al., 2011), a corpus composed of 1,537,283 Flemish Dutch Netlog posts is presented, labeled with the age and gender of the authors. Some of the preprocessing that is performed consists in the removal of quotes from previous posts, the interpretation of emoticons and the normalization of the tokens that contain four or more consecutive identical characters to three (changing e.g., *niiiice* to *niiice*).

3.1.1.5 Movie Reviews

Movie reviews have also been used in author profiling. Otterbacher (2010) collects a dataset containing movie reviews from IMDB[5] labeled by the gender of the authors. The comments that are crawled correspond to the reviews of the 250 top films of all times according to the website.

3.1.1.6 Chat Logs

Another type of texts used to perform gender identification are chat logs. Kucukyilmaz et al. (2006) retrieved around 250,000 chat logs written in Turkish from 1,500 users of a chat server (Heaven BBS) and classified these messages with respect to the gender of their author. The messages from 1,500 users are considered; 50,000 distinct words are used in the corpus; the mean number of words per message is 6.2. The style of chat messages is quite different from any type of textual data. Some of the characteristics that make chat messages special are the following: the use of punctuation marks varies widely for each user (some users omit punctuation marks in their messages while others overuse them), emoticons are a very important part of the messages, and misspellings occur frequently, not only where the user commits orthographic errors, but also where the users put emphasis on an expression (e.g., *awesomeeeee*).

3.1.1.7 Student Essays

Student essays are often used in author profiling. This kind of text is often used in a specific task: native language identification (in few cases, student essays are

[4] `https://www.twoo.com/`
[5] `http://www.imdb.com/`

used in personality identification as well). The goal of this task is, given texts written in a specific language, to predict the mother tongue of the writer. Two of the most used resources in native language identification are "The International Corpus of Learner English" (ICLE) and the TOEFL11 corpus, both of which are composed of texts written by language learners.

ICLE

This corpus contains argumentative essays written by higher intermediate to advanced learners of English with several mother tongue backgrounds: Bulgarian, Chinese, Czech, Dutch, Finnish, French, German, Italian, Japanese, Norwegian, Polish, Russian, Spanish, Swedish, Tswana, and Turkish. It contains of 6,085 essays and 3.7 million words.

Tetreault and Blanchard (2012) analyze the characteristics of ICLE and address the problems that this corpus brings to the task of native language identification, such as its small size and topic bias. As a response to these problems, the authors introduce the TOEFL11 corpus.

TOEFL11

This dataset contains 12,100 essays that correspond to the responses provided by test takers of the TOEFL test[6] in 2006. The essays are written by native speakers of Arabic, Chinese, French, German, Hindi, Italian, Japanese, Korean, Spanish, Telugu, and Turkish. The corpus contains 1,100 essays per language sampled evenly from eight different topics. The score of each essay, given by the teacher to the writer of the essay, is also provided.

In (Argamon et al., 2005), the authors use a corpus composed of essays written by students of the university of Texas between 1997 and 2003. The authors were given scores for the five personality factors; subjects with scores in the top third of each of the studied traits were labeled as high, and the ones in the bottom third as low.

3.1.1.8 News Corpora

News datasets have also been used. Tofighi et al. (2012) collected texts from news agencies where English, Persian, Turkish, and German native authors write articles in English. 150 texts by authors of each native language are collected and used in the study.

[6] `https://www.ets.org/toefl`

"The Reuters Corpus Volume 1" (Lewis et al., 2004) is another news corpus that has been used in the field. It consists of all English language stories produced by Reuters' journalists between August 20, 1996 and August 19, 1997 and has been manually annotated by the gender of the author.

3.1.1.9 Personal Ads

Groom and Pennebaker (2005) compiled a set of Internet personal advertisements placed by heterosexual and homosexual men/women. This type of texts offers an unambiguous operational definition of sexual orientation: the ads state both the sex of the ad-poster and the sex of the desired mate. The chosen site for the study is Match.com, which has more than three million members. 4,000 ads from each of the four mate-preference groups have been retrieved. Advertisements are selected within each category, copied, and pasted into a spreadsheet in the order in which they appeared. The ads which were obviously misplaced (with respect to sexual orientation) were manually excluded.

3.1.1.10 Political Texts

Makazhanov and Rafiei (2013) use a collection of politically relevant tweets, defined as tweets that contain at least one politically relevant hashtag. Tweets of 1,000 users were labeled manually with three categories: "left", "right" and "ambiguous".

Kapovciute-Dzikiene et al. (2015) use text transcripts of the Lithuanian parliamentary speeches and debates. Three datasets were compiled, each one labeled by the age, gender and political alignment of the speakers.

3.1.1.11 Scientific Articles

Sarawgi et al. (2011) collected papers from researchers of the Natural Language Processing community. The authors randomly selected 5 female and 5 male authors and retrieved 20 papers from each of them. The papers are about different subtopics of NLP. The first 450 words, preserving the sentence boundaries, are used in the corpus. Each instance is labeled by the gender of the author manually (in the cases of multiple authors, the gender of the most senior author is used).

3.1.2 Approaches to Author Profiling

In this section, we analyze different author profiling approaches. We organize these approaches depending on the demographic traits that are being classified and the kind of data that is used.

3.1.2.1 Gender and Age Identification

The most often explored demographic traits in the literature are clearly gender and age.

Using Blog Data

In (Schler et al., 2006), the authors describe two experiments on The Blog Authorship Corpus: gender and age identification. For both experiments, stylistic features (function words, part-of-speech frequencies, blog words and hyperlinks) as well as the 1,000 most relevant unigrams according to the Information Gain metric are extracted. The chosen classifier is the Multi-class Real Winnow, which achieves 80.1% accuracy for gender and 76.1% for age identification.

The same dataset is used in (Goswami et al., 2009), where a stylometric analysis of the age and gender of the bloggers is presented. Two main novel features are used, sentence length variation and non-dictionary words. The motivation for the second group of features is that after analyzing the word usage per gender and age, the authors concluded that teenagers generally use more non-dictionary words than adults. The final feature set contains the average sentence length, the frequencies of 35 content words as well as 52 slang words. There is an improvement in terms of performance compared to previous approaches that use the same data, obtaining 89.18% in gender and 80.32% in age identification with Naïve Bayes as classifier. Both this approach and the previous one are very content dependent, their systems depend mainly on the specific word choices of the authors. The main difference is the dimensionality of the feature vectors: while Schler et al. (2006) use more than 1,000 features, Goswami et al. (2009) use less than 100, obtaining better results.

In (Nguyen et al., 2011), the goal is to identify the age of the authors using three different datasets: The Blog Authorship Corpus (Schler et al., 2006), The Fisher Corpus (David et al., 2004), and a collection of posts from a breast cancer forum. A linear regression model is used and each feature is represented by a vector which contains the value of that particular feature globally and for each dataset. The feature set is composed of token unigrams, part-of-speech unigrams and bigrams; further features that are extracted using the LIWC tool (Pennebaker et al., 2001) are also considered. These LIWC features consist of the frequencies of inclusion words (e.g., *with*, *and*, *include*, etc.), causation words (e.g., *because*, *hence*, etc.), and stylistic characteristics such as percentage of words longer than 6 letters. The gender of the authors/speakers is also used as a feature. The authors achieve accuracies of 69.9% in the blog data (which is lower than what the two previous approaches commented above achieved on this dataset), 74.2% on the telephone transcriptions, and 53.5% in the breast cancer forum posts. An analysis

of the most important features per age group (old, young) is presented: words such as “like”, “mom” or “definitely” characterize younger authors, and words such as “years”, “daughter” or “grandson” are characteristic of older authors.

Argamon et al. (2009) present four different experiments in which gender (“man” vs “woman”), age (13-17, 23-27, 33-47), native language (Russian, Czech, Bulgarian, French, and Spanish), and personality (“neurotic” vs “non-neurotic”) of the authors are predicted. To do so, two kinds of features are extracted: style-based features and content-based features. The stylistic features are computed by using taxonomies provided by systemic functional linguistics (Halliday and Matthiessen, 2004), which describe meaningful distinctions of function words and parts of speech. The content-based features consist of the 1,000 words that have the highest Information Gain coefficient in the training set. Three different datasets are used: The Blog Authorship Corpus (Schler et al., 2006) is used for the gender and age experiments; the ICLE for the native language identification experiment; and a set of essays written by psychology undergraduates for personality classification. To classify, the authors use a Bayesian Multinomial Regression. Their results are presented in Table 3.1.

	Baseline	**Style**	**Content**	**Style+Content**
Gender (2 classes)	50.0	72.0	75.1	**76.1**
Age (3 classes)	42.7	66.9	75.5	**77.7**
Language (5 classes)	20.0	65.1	**82.3**	79.3
Neuroticism (2 classes)	50.0	**65.7**	53.0	63.1

Table 3.1: Results of Argamon et al. (2009).

As shown in Table 3.1, the reported results on gender identification are the same as in (Schler et al., 2006).

Gender classification is further explored in (Mukherjee and Liu, 2010). A new blog post corpus is crawled for this work. F-measure (Heylighen and Dewaele, 2002), stylistic features, gender preferential features, factor analysis, and word classes as well as part-of-speech sequence patterns are used as features. The authors present a new ensemble feature selection algorithm to select the most discriminative features. The stylistic features that are considered are: parts of speech, unigrams and words specific to blogs. Ten different features are extracted that analyze the suffixes of words; the authors call this group of features *gender preferential features*. Factor analysis and word classes refer to dictionaries of words that have positive/negative or emotional content and dictionaries that include words that refer to the topics “conversation”, “home”, “family”, “food/clothes” and “romance”. Part-of-speech sequence patterns are sequences of consecutive part-of-speech tags that satisfy predefined constraints. The best reported result is 88.56%

accuracy, achieved using a Support Vector Machines classification algorithm after applying the feature selection algorithm.

Sarawgi et al. (2011) present a gender and genre identification model that avoids gender bias in topics. As corpus, blog posts (a corpus specifically crawled for this work) and a collection of scientific papers are used. Three groups of features are used: deep long-distance syntactic patterns based on probabilistic context-free grammars (PCFGs), token-level language models, and character-level language models. As baselines, a simple bag-of-words approach and two publicly available systems[7] are drawn upon. The accuracies of gender identification in the blog dataset are presented both by topic and in average. The mean accuracy of 68.3% with the character-based model in the cross-topic scenario is the best result. In the scientific domain, character-based features and syntactic features perform equally, achieving 76% of accuracy. In another presented experiment, the system is trained on the blog dataset and tested on the scientific dataset; the accuracy in this experiment decreases significantly due to the differences of the texts used in each phase. Different techniques are used to classify the instances: when using PCFGs, the classification method consists in computing the similarity between the PCFG of the test document and the PCFGs that represent each gender. The token and character-level language models are classified using the LingPipe package[8]. The bag-of-words baseline uses the Maximum Entropy classifier.

Rosenthal and McKeown (2011) aim to predict the age of blog post authors. A collection of texts from the blog platform LiveJournal is compiled for this work. Three experiments are performed: classification of the authors into three age groups, binary classification between blog posts by authors born before or after each year between 1975-1988, and a more detailed classification which takes a closer look at the authors born in 1979 and 1984, using different sets of features. The goal of this study is to analyze whether the emergence of social media technologies produced a shift in the writing style among college-aged students in that generation. The feature set is composed of three feature groups: online behaviour and interest, lexical content, and stylistic features. The first group contains metadata of the profile page of the author such as the listed interests, number of friends, posts, average number of comments, etc. The lexical[9] content feature group contains the frequencies of emoticons, acronyms, slang words, punctuation marks, capitalizations, sentence lengths and links/images. Finally, the last group contains part-of-speech bigrams of each age group and the 200 most frequent words. Logistic regression is used to predict the age. The first experiment achieves an ac-

[7] gender guesser: `http://www.hackerfactor.com/GenderGuesser.php` and gender genie (no longer available)

[8] Available at `http://alias-i.com/lingpipe/`

[9] Even though we disagree on the lexical nature of some of these features, we use their terminology.

curacy of 67%. In the second experiment, the authors conclude that content helps more than style, but style helps more as age decreases. In the last experiment, their best performances achieve 79.96% for 1979 and 81.57% for the year 1984.

Another work that performs several classification tasks using informal blog posts is (Pham et al., 2009). In this case, the chosen language is Vietnamese rather than English. 298 features compose the feature set. The features can be grouped into word-based, character-based, function words, structural, line-based, paragraph-based, lexicon-based, content-specific, and part-of-speech features. The baseline is the majority class classifier. The performance of the system is shown in Table 3.2.

Trait	ML Algorithm	Features	Feature Sel.	Baseline	Result	Improvement
Age	IBk (IB1)	all	None	45.80	77.27	+21.47 (47.1%)
Location	IBk (IB1)	all	None	44.15	78.01	+33.86 (76.7%)
Gender	IBk (IB1)	all	None	59.90	83.34	+23.44 (39.1%)
Occupation	Random Forests	all	None	57.23	82.12	+24.89 (43.5%)

Table 3.2: Results of Pham et al. (2009).

In the table, it is shown that the results of each experiment outperform the baseline by a large margin. Each experiment is performed using the full feature set and without applying feature selection techniques. The best performing machine learning algorithms are IBk (Weka's implementation of K Nearest Neighbors) and Random Forests.

Using Generic Reference Corpora

In (Koppel et al., 2003), the authors use the British National Corpus (Clear, 1993) to distinguish between male and female authors as well as the genre of the text (fiction vs non-fiction). Part-of-speech frequencies and function words are used as features. The learning method is a variant of the exponential gradient algorithm. The system obtains accuracies of 77.3% in genre distinction and 79.5%-82.6% in gender identification, depending on what genre is used to train the classifier. The authors also experiment with feature reduction algorithms, analyzing the performance of the system with respect to the number of features.

Using Email Data

In (Cheng et al., 2009), the authors use the Enron email corpus. To perform gender identification on emails, the authors extract five subsets of features: character-based (e.g., upper-cased characters, usage of white-spaces, etc.), word-based (e.g., vocabulary richness, number of short words, chars per word, etc.),

syntax-based[10] (usage of quotes, commas, periods, colons, semi-colons, question marks, exclamation marks, and ellipsis), structure-based (e.g., number of lines, sentences per line, number of paragraphs, etc.), and function words. The total number of features is 545. SVM is used to classify, achieving 82.20% accuracy in the best case.

An extension of this work is presented in (Cheng et al., 2011), where the same feature set is used in two scenarios. The first one was already presented in (Cheng et al., 2009). The second one uses the Reuters Corpus. The experiments predict the gender of the author in both scenarios, obtaining the same accuracy in the first case as in their previous work. In the case of the Reuters Corpus, the authors present an accuracy of 76.75% when SVM is used for classification. The authors conclude that predicting the gender of the author of neutral news is a much more challenging problem than in the case of personal emails.

Another approach that uses email messages (but does not use the Enron corpus) is described in (Estival et al., 2007). 689 features are extracted. The feature set includes character-level features (e.g., punctuation, word length, case-based features, etc.), lexical features (e.g., function words, parts of speech, named entities, etc.) and structural features (e.g., paragraph breaks, presence of some HTML tags, presence of cited text from previous emails, etc.). The results of each classification experiment are shown in Table 3.3.

Trait	**ML algorithm**	**Feature Sel.**	**Best Features**	**Results**	**Baseline**
Age	SMO	-	all	56.46	39.43
Gender	SMO	-	all	69.26	54.48
Language	Random Forests	InfoGain	all-correlate	84.22	62.90
Education	Bagging	-	all-functionWord	79.92	58.78
Country	SMO	-	all	81.13	57.29
Agreeableness	IBk	-	char + structural	53.16	40.51
Conscientiousness	IBk	-	char + structural	54.35	43.72
Extraversion	LibSVM	-	char + structural	56.73	45.17
Neuroticism	IBk	-	char + structural	54.29	42.34
Openness	Random Forests	-	structural	55.32	47.28

Table 3.3: Results of Estival et al. (2007).

In the table, it is shown that every experiment outperforms the baseline (majority class classifier). The best improvement over the baseline is obtained in the country identification experiment. The personality trait classification experiments show worse results and smaller improvements over the baseline. The best performing classifiers are SMO (a SVM variant), Random Forests, Bagging, IBk and LibSVM.

[10]Even though we don't consider this group of features of syntactic nature, we use their terminology.

Using Twitter Data

Burger et al. (2011) present a gender identification approach that uses Twitter data. The selected feature set consists of character 1-5grams and word 1-2grams from the content of the tweets and screen name, full name and description of the profile. This generates more than 15 million distinct features, which presents a challenge to most machine learning toolkits. Using the Balanced Winnow algorithm and some code optimizations, 92% of accuracy is achieved when all the previously mentioned features are combined. The system is also tested using only the content of the tweets. This approach performs worse, obtaining 76% of accuracy. Both alternatives outperform a human classification approach, implemented with Amazon Mechanical Turk[11] workers, who were asked to do the classification manually.

Twitter data is also used in age identification experiments. In (Nguyen et al., 2013), a collection of tweets written in Dutch is compiled and used in three different experiments: classification using age categories as labels, prediction of the exact age of the authors and classification of the users by their *life stage* (secondary school student, college student, employee). In each experiment, the gender of the author is also taken into account. The chosen feature set is composed of unigrams that occur at least ten times in the training set. Logistic regression is used to predict the age. The authors compare their automatic system with a manual approach and prove that the implemented system is much more effective. Results show F1 measure values of 76.70% in the age category classification and 67.85% in the life stage experiment. An analysis on how feature values change with age is also presented. The authors conclude that style-based features such as the vocabulary complexity or word capitalization vary significantly with age.

Using Facebook Data

As already discussed in the section on the datasets, the social network Facebook has also been used in this line of research. In (Rangel and Rosso, 2013b), a cognitive approach based on neurology studies is presented. The goal is to classify the authors of texts by their gender and age. A study on the frequencies of each grammatical category in 6 different sources (namely Wikipedia, newsletters, forums, blogs, Twitter and Facebook) is outlined. Focusing on Facebook data, the same analysis for each gender is also performed, concluding that in Spanish, men use more prepositions than women, and that on the other hand, women use more pronouns, determiners and interjections. After these remarks, a feature set is presented, composed of word-based features (such as the words that start with

[11] `https://www.mturk.com/mturk/`

a capital letter, words with all characters capitalized, word length, etc.), usage of punctuation marks, frequency of each part of speech, number of emoticons and the usage of emotion words (using the Spanish emotion lexicon introduced in (Sidorov et al., 2013)). The experiments use the "PAN Author Profiling Task 13" Spanish data (Rangel et al., 2013) and SVM for classification. The results show competitive performance compared to the other participants of the task.

The same authors perform gender identification as well as emotion detection using Facebook comments in (Rangel and Rosso, 2013a). The feature set is the same as described in (Rangel and Rosso, 2013b). Both gender identification and emotion detection tasks are casted as binary classification problems. In the first case, the classifier distinguishes between male and female authors and in the latter, for each emotion, the classifier determines whether the text expresses this emotion or not. Four different classifiers are tested, namely Weka's (Hall et al., 2009) implementation of J48, Naïve Bayes, SVM and Bayes Nets. The gender identification experiment obtains an accuracy of 59%. The emotion detection experiment, on the other hand, obtains variable results, depending on each emotion: 59.6% for joy, 32.3% for anger, 36.1% for disgust, 50.4% for surprise and 20% for sadness.

Using Netlog Data

In (Peersman et al., 2011), the authors use content generated in the Belgian social network Netlog[12]. The chosen set of features consists of the 1,000, 5,000, 10,000 and 50,000 most informative (according to the chi squared coefficient) token unigrams, bigrams and trigrams as well as character bigrams, trigrams and tetragrams. The experiments distinguish between the following labels: "<16" vs ">16", "<16" vs ">18", "<16" vs ">25" and "<16male" vs "<16female" vs ">25male" vs ">25female". An SVM classifier is used for classification. The first experiment classifies texts based on age range (whether the author is younger or older than 16 years of age). The system achieves 71.3% of accuracy, which improves as the age gap increases, rising to 80.8% when classifying authors younger than 16 vs older than 18, and to 88.2% when authors younger than 16 vs older than 25 are considered. The last experiment differentiates between males and females younger than 16 and males and females older than 25. The best result achieves 66.3% of accuracy using the 50,000 most distinctive token unigrams, which are the best performing features in all experiments.

[12] https://www.twoo.com/

Using Movie Reviews

Otterbacher (2010) uses a dataset of movie reviews from IMDB[13] labeled by the gender of the authors. To perform gender identification, a statistical regression model is used. Three groups of features are used: aspects of author writing style, content of the review, and metadata of the reviewed movie and of the review. The writing style features are the vocabulary richness, a list of the fifty most common words, text complexity, a list of 55 hedges (expressions such as "kind of", "sort of" used to soften an argument. Their list of hedges is based on (Lakoff, 1973)), and the usage of pronouns. From the content of the review, measures of centrality, perplexity, entropy and out of vocabulary rate are extracted. The author also extracts twenty groups of lexical items by constructing a semantic space on which latent semantic analysis is performed. The metadata features use meta-information such as the length of the review, the reviewer rating and the movie popularity. Combining all the mentioned features, the gender of the authors is predicted correctly in 73.71% of the cases.

Using Chat Messages

Kucukyilmaz et al. (2006) use both term-based and style-based features (e.g, usage of emoticons, stopwords, punctuation, character usage, etc.) to perform gender identification on chat messages. The best results are achieved using Naïve Bayes with the style-based features, resulting in 81.7% of accuracy. The effects of feature selection are also studied, and a list of the most distinctive words (with respect to the chi squared coefficient) is presented. More information on the usage of chat logs in the field of author identification and the applications of those approaches to forensic studies and cybersecurity can be found in Section 3.2.

3.1.2.2 Native Language Identification

Another demographic trait that has been explored in author profiling is native language. The experiments that are often implemented aim to predict the mother tongue of the author of a text.

Using Language Learner Essays

A native language identification approach is presented in (Koppel et al., 2005). The goal of the study is to determine whether the mother tongue of the authors is

[13] `http://www.imdb.com/`

Czech, French, Bulgarian, Russian, or Spanish, using student essays written in English from the ICLE corpus. To do so, three feature types are extracted: function words, letter n-grams and errors and idiosyncrasies. The goal of the last group of features is to determine whether the writer transports orthographic or syntactic conventions from his/her native language over to English in non-conventional ways. The error types that are considered are orthographic, syntactic, neologisms and part-of-speech bigrams rarely used in standard English. The authors develop automated methods to recognize these kinds of errors. The final set of features used is the following: 400 standard function words, 200 character n-grams, 185 error types, and 250 rare part-of-speech bigrams. To classify, a multi-class linear SVM is used. Using the whole set of features, 80.2% of the texts are correctly classified. The errors are manually analyzed and some insightful conclusions are extracted, such as that Spanish and Czech authors had difficulty doubling consonants, Russian students are more prone to use the word "over", and the frequency of the word "the" is much lower in the writings of Czech, Russian and Bulgarian authors than in those of French and Spanish authors. The low frequency of the word "the" is caused by the native language characteristics of the authors (Russian and Czech do not have articles, and in Bulgarian definiteness/indefiniteness is marked by a suffix).

Another native language identification approach is (Wong and Dras, 2009). The authors also use the ICLE corpus. The same mother tongues as in (Koppel et al., 2005): Czech, French, Bulgarian, Russian and Spanish, as well as Chinese and Japanese, are chosen. Syntactic features are used. These features measure three major syntactic error types: subject-verb disagreement in terms of number or person, noun-number disagreement and misuse of determiners and pronouns. These features are combined with the features used in (Henderson et al., 2013): character n-grams, part-of-speech n-grams and function words. The combination of features predicts the native language correctly in 73.71% of the cases. SVM is used for classification. In the presented experiments, using syntactic errors as features does not outperform standard features, arguably as a result of the small number of syntactic error types being considered.

Tetreault and Blanchard (2012) present a native language identification system applied to different datasets and a native language identification corpus composed of student essays from the TOEFL exam. The four used datasets are: the ICLE corpus, a subset of the TOEFL corpus that contains texts written by authors with the same seven native languages considered in the ICLE (refered to as "TOEFL7"), a subset of the corpus that contains essays of each native language (referred to as "TOEFL11"), and the full corpus (referred to as "TOEFL11-Big"). The feature set is composed of character n-grams, function words, part-of-speech bigrams, spelling errors, word n-grams, writing quality features, tree substitution grammar fragments, Stanford dependencies, and perplexity scores from 5-

gram language models. A logistic regression ensemble of every feature set is implemented. Accuracies of 90.1% in the ICLE, 70.9% in TOEFL7, 80.9% in TOEFL11, and 84.6% in TOEFL11-Big are reported. The authors also try to train the system with the ICLE and to test with TOEFL7, which performs poorly, but, on the other hand, the inverse process obtains 67.4% of accuracy. The drawn conclusion is that training on a larger corpus and testing on a smaller one works reasonably well, however training on a small corpus and testing on a larger one, does not yield good results with the used feature set.

In (Henderson et al., 2013), the authors use the TOEFL11 data. The features that are used were presented in (Burger et al., 2011). An ensemble of three classifiers is used. The ensemble is composed of an algorithm developed by the authors called *Carnie*, Liblinear[14] and the SRI Language Modeling Toolkit [15]. The problem is casted as an 11-class (each class being the native language of the student) classification problem. 82.6% of accuracy is achieved by the ensemble, outperforming the approach by Tetreault and Blanchard (2012) by more than 2%.

Wong and Dras (2011) use the ICLE dataset. Three basic feature groups are taken into account: lexical features (the same feature set presented in (Henderson et al., 2013)), production rule features, which are horizontal slices of parse trees and part-of-speech n-grams. To obtain the parse trees, two parsers are used, *The Stanford Parser* (Klein and Manning, 2003), and *The Charniak and Johnson Parser* (Charniak and Johnson, 2005). A maximum entropy machine learning algorithm is used to classify. The classifier achieves 81.71% of accuracy, which is lower than the reported results of Tetreault and Blanchard (2012) on the same data. The authors conclude that using sections of parse trees improves the lexical model for native language identification. An analysis of the types of syntactic substructures that are useful for classification is also provided.

Using News Articles

Tofighi et al. (2012) use news texts where English, Persian, Turkish and German native authors write articles in English. A collection of 386 features is used. This feature set can be divided into: lexical features (e.g., number of characters, number of tab spaces, number of upper characters, number of short words, vocabulary richness etc.), syntactic features (e.g., number of quotes[16], ellipsis, etc.), frequencies of a list of 300 function words and structural features (e.g., number of lines, sentences, paragraphs, greeting words, farewell words, etc.). Three classification techniques are used: SVM, Naïve Bayes, and C4.5. The best classification

[14] `https://www.csie.ntu.edu.tw/~cjlin/liblinear/`

[15] `http://www.speech.sri.com/projects/srilm/`

[16] Even though we do not consider this feature of syntactic nature, we use their terminology.

result is obtained by SVM, which achieves an accuracy of 86.44%.

3.1.2.3 Personality Identification

Author personality is another trait that has been explored. The approaches that attempt to characterize the personality of an author usually rely on the Big Five personality characteristics: openness, conscientiousness, extraversion, agreeableness and neuroticism. For more information on the development and history of the Big Five personality trait theory, see (Goldberg, 1993).

Using Student Essays

In (Argamon et al., 2005), the authors focus on two personality dimensions, namely extraversion and neuroticism. The authors perform binary classification experiments that predict whether an author of an essay has high or low values of extraversion and neuroticism. The used feature set is composed of function word frequency, features based on systemic functional grammars, which model language in terms of interpersonal, textual and propositional functions, and text cohesion, assessment and appraisal measures. SVM is used to perform classification. The authors observe that the most useful features to measure neuroticism are the appraisal metrics, which predict correctly 58.2% of the neuroticism levels of the authors. The extraversion experiments obtain 58% of accuracy, using the full feature set. The most informative set of features in this case are the frequencies of function words.

A different approach is presented in (Poria et al., 2013a). The goal is the same as in the previously reviewed approaches: given a text, to determine whether each of the personality traits has high or low score. Student essays are used as input data. Several resources are used to compile the feature set: namely the LIWC, which is used to extract 81 features related to the word count frequency of different word categories, and MRC (Wilson, 1988), which is a database of psycholinguistic categories, and a machine usable dictionary used to compute features such as the number of phonemes/syllables/letters in a word or the stress pattern. The authors furthermore use emotional features extracted from SenticNet (Cambria et al., 2012)[17], ConceptNet (Havasi et al., 2007)[18] and EmoSenticNet (Poria et al., 2013b)[19]. The combination of the mentioned features with the SMO classifier

[17]A lexical resource that contains concepts along with polarity scores, including 7,600 multiword concepts.

[18]Conceptnet represents the information from the open mind corpus (Chklovski and Mihalcea, 2002) as a directed graph where nodes are concepts and edges are common-sense assertions that connect them.

[19]EmoSenticNet contains about 5,700 common-sense knowledge concepts.

produces the results displayed in Table 3.4.

Trait	Precision	Recall	F-score
Openness	0.662	0.662	0.661
Conscientiousness	0.634	0.634	0.633
Extraversion	0.636	0.636	0.634
Agreeableness	0.622	0.622	0.615
Neuroticism	0.637	0.637	0.637

Table 3.4: Results of Poria et al. (2013a).

The Big Five personality theory is also used in (Verhoeven et al., 2013). The authors create an ensemble meta learner composed of five classifiers that perform 10-fold cross validation using Facebook training data and five more on essay data (one for each personality trait). All classifiers used in the ensemble are standard SVMs. The feature set is composed of the 2,000 most frequent character trigrams and the outputs of the classifier ensemble. The first experiment shows the accuracies of each one of the components of the ensemble individually on a 10-fold cross validation scenario using all the data. It is shown that the performance is better on Facebook data than on the essays (obtaining values of 50-56% accuracy in the essay data and 61-75% in the Facebook data). Using the ensemble for training and testing on the Facebook test data, the performance improves, especially for the openness and the conscientiousness traits.

There have been also approaches in which the standard big five personality traits are not used to characterize the authors' personality. Luyckx and Daelemans (2008c) follow the Myers-Briggs Type Indicator (Myers and Myers, 2010), which categorizes a person with respect to 8 opposed factors: "introversion", "extraversion", "intuition", "sensing", "feeling", "thinking", "judging", and "perceiving". n-grams of both fine-grained and coarse-grained part-of-speech tags are used as features. Eight binary classification tasks are performed, with the goal to assign to the authors one or several of the above factors. The results of the experiments show that the system identifies the "judging" authors with higher accuracy than the other factors. In another experiment, the authors are classified in four binary classification processes to distinguish between "introverts" vs "extroverts", "intuitives" vs "sensers", "feelers" vs "thinkers" and "judgers" vs "perceivers". The subtask that leads to best results is the classification between "'judgers" and "perceivers", where the classifier is able to predict correctly 82.07% of the cases.

Using Blog Posts

Oberlander and Nowson (2006) use a corpus of personal weblog posts. The features that are extracted are different selections of n-grams. Several splits of the data are considered. Two different experiments are shown. The first one aims to distinguish between low and high levels of the considered personality traits (neuroticism, extraversion, agreeableness and conscientiousness). The second experiment is cast as a multi-class classification problem that distinguishes between highest, relatively high, medium, relatively low and lowest values of each personality trait. Naïve Bayes and SVM are used for classification. Depending on the data split and the selected n-grams, a different accuracy is presented for each of the traits, obtaining as maximum performance 96% for neuroticism and 100% for the remaining traits for the binary classification tasks, and 94.4% for neuroticism, 90.1% for extraversion, 90.4% for agreeableness and 92.3% for conscientiousness in the multi-class task.

3.1.2.4 Sexual Orientation Identification

A trait that has not been studied thoroughly in the state of the art of author profiling, is sexual orientation. Groom and Pennebaker (2005) aim to assess whether sexual orientation influences linguistic behavior or not. More precisely, the goal is to see whether popular stereotypes of the LGBT community (lesbian women try to be like men, and gay men try to be like women) can be detected at the linguistic level. The LIWC software is used to analyze the frequencies of different word categories. The results of the study provide evidence that gay men and lesbians show patterns that are similar to heterosexual men and women respectively, a fact that undermines the assumptions that predict gender-atypical linguistic characteristics in gay people.

3.1.2.5 Political Alignment Detection

A characteristic that cannot be classified as a demographic trait but that has been also studied using similar methodology, is political alignment.

Makazhanov and Rafiei (2013) extract content and network-level features from tweets. The content related features consist of the most relevant unigrams (in terms of the tf-idf score) and the frequency with which a user uses a specific hashtag. Network-level features focus on the relations between users instead of focusing on the content of the tweets themselves (relying on the homophilic properties of the social network: users prefer to connect to those who are like themselves). SVM is used to classify. Using unigrams, the system achieves 79% of accuracy, while hashtags predict correctly 90.8% of the cases. Clustering the users using the

network-level features groups correctly users with the same political alignments in 95% of the cases. Combining both groups of features does not outperform the performance of the network model.

Similar to the previous approach, Kapovciute-Dzikiene et al. (2015) perform gender, age and political alignment identification using text transcripts of the Lithuanian parliament. Style markers, character n-grams, function words, word n-grams, lemma n-grams and part-of-speech n-grams compose the feature set. SVM is used to classify the speakers in terms of gender, age and political alignment. The best reported results predict correctly 58.7% of the cases in the political alignment, 74.6% in gender determination, and 44.6% in age determination tasks. Lemma n-grams are the most distinctive features in each of the classification tasks.

3.1.2.6 Sociolinguistic and Psychological Studies

Studies that analyze language from sociolinguistic and psychological perspectives need to be mentioned as well. Consider, e.g., (Pennebaker et al., 2003) for a study on how the words people use in their daily lives reveal important aspects of their social and psychological worlds; (Nerbonne, 2013) for a study that analyzes pronoun usage and tries to relate it to the writers' demographic characteristics; and (Slatcher et al., 2007) for a study on the individual differences in linguistic style among U.S. presidential and vice presidential candidates. For further analysis of the influence of gender in language and a collection of articles that examine the dynamic ways in which women and men develop and manage gendered identities through their words, see (Holmes and Meyerhoff, 2008).

3.1.3 Shared Task

PAN is an author profiling shared task organized every year. The shared task provides texts labeled by the gender and age of the authors, written in English and Spanish (with the recent addition of Dutch) extracted from Twitter and other social networks. The participants of the task deploy author profiling systems that predict both the gender and age of the authors (and personality traits in the last edition). Overview papers of the task that describe both the provided data and the submissions are (Rangel et al., 2013, 2014b, 2015, 2016).

3.1.4 Summary

To summarize the author profiling state-of-the-art review, we first provide Table 3.5, which contains some of the publicly available resources used in several approaches.

Table 3.5 shows a clear tendency of using in the task of author profiling, informal texts and social media, which present the additional problem of noise as an extra challenge that is to be solved. The table also reveals the preference of focusing on gender and age as favourite predicted demographic traits. Then, a summary of the different approaches applied to author profiling is shown in Table 3.6.

After reviewing the related work on author profiling, several conclusions can be drawn. The selected features tend to rely heavily on the content of the texts, which limits their potential cross-domain adaptability. Content-based approaches tend to use thousands of lexical features (token/character n-grams, frequent words, etc.) for classification. This can be very effective for cases where large amounts of written material in the same domain and sufficient computational power are available. We believe that this sort of tasks can be solved in a more optimal way, namely by extracting deeper linguistic features that exploit the profiling potential of syntax and discourse.

When syntactic features are mentioned, often function words and punctuation marks are meant, which is an oversimplification and a very shallow approach to syntactic feature extraction; see e.g., (Amuchi et al., 2012; Abbasi and Chen, 2005; Cheng et al., 2009).

ID	Resource Name	Size	Language	Labeling	Source	Reference
BAC	The Blog Authorship Corpus	>680K texts	EN	Gender Age Occupation Astro. sign	Blogger Posts	(Schler et al., 2006)
FC	The Fisher Corpus	16K texts	EN	Gender Age	Telephone Transcripts	(David et al., 2004)
BNC	The British National Corpus	4049 texts	EN	Gender, Age Author Geographic origin	Formal Texts	(Clear, 1993)
P13AP	PAN13 author profiling task corpus	>300K texts	EN, ES	Gender, Age	Informal Blog Posts	(Rangel et al., 2013)
P14AP	PAN14 author profiling task corpus	>8800 texts	EN, ES	Gender, Age	Tweets, Hotel Reviews	(Rangel et al., 2014b)
P15AP	PAN15 author profiling task corpus	Not Specified	EN, ES, NL, IT	Gender, Age Personality	Tweets	(Rangel et al., 2015)
ICLE	International Corpus of Learner English	3.640 texts	EN	Native Language	Student Essays	(Granger, 2003)
TOEFL	TOEFL Corpus	87.502 texts	EN	Native Language	Student Essays	(Tetreault and Blanchard, 2012)
REUT	Reuters Corpus Volume 1	> 800K texts	EN	Gender	News Articles	(Lewis et al., 2004)
FBPI	Facebook Personality and Irony Corpus	1200 texts	ES	Personality Traits and Irony	Facebook Posts	(Rangel et al., 2014a)
NLC	Netlog Corpus	>1.5M texts	NL	Gender and Age	Netlog Posts	(Peersman et al., 2011)
IMDB	IMDB CORPUS	31.300 texts	EN	Gender	IMDB Reviews	(Otterbacher, 2010)

Table 3.5: Publicly available resources for author profiling.

Reference	Labels	Features	Corpus	Accuracy	Classifier
(Schler et al., 2006)	Gender and Age	Unigrams, function words, PoS, links, blog words	Blogs	80.1% Gender 76.1% Age	Winnow
(Goswami et al., 2009)	Gender and Age	Sentence length, content words, slang	Blogs	89.18% Gender, 80.32% Age	naïve Bayes
(Nguyen et al., 2011)	Age	Unigrams, PoS, bigrams and LIWC	Blogs, Telephone transcripts and forum entries	69.9% blogs, 74.2% transcripts. 53.5% forum	Linear Regression
(Argamon et al., 2009)	Gender, Age, Native Lang and Personality	Taxonomies 1000 most distinctive tokens	Blogs and student essays	76.1% Gender, 77.7% Age, 82.3% NL, 65.7% pers	Bayesian Multinomial Regression
(Mukherjee and Liu, 2010)	Gender	PoS, unigrams, blog words, ending of words, dictionaries	Blogs	88.56%	SVM
(Sarawgi et al., 2011)	Gender	PCFG patterns, word/char language models	Blogs and scientific articles	68.3% in blogs, 76% scientific articles	Similarity between PCFGs and MaxEnt
(Rosenthal and McKeown, 2011)	Age	Meta-data, emoticons, acronyms, slang, punctuation, frequent words, PoS collocations	Blogs	67%, 79.96% and 81.57%	Logistic Regression
(Pham et al., 2009)	Age, Location, Gender, Occupation	Word/char-based, function words, line/paragraph based and structural	Blogs	77.27% age, 78.01% location, 83.34% gender, 82.12% occupation	IBK and Random Forests
(Koppel et al., 2003)	Gender and Genre	PoS and function words	Formal Texts	77.3% Genre, 82.6% Gender	Winnow
(Cheng et al., 2009)	Gender	Char/word based, structure based, syntactic and function words	Emails	82.20%	SVM
(Cheng et al., 2011)	Gender	Char/word based, structure based, syntactic and function words	Emails and News	82.20% emails, 76.75% news	SVM
(Estival et al., 2007)	Age, Gender, Language, Education, Country, Personality	Punctuation marks, word length, function words, named entities, PoS, structural features	Emails	56.46% age, 69.26% gender, 84.22% language, 79.92% education, 81.13% country and 53.16%-56.73% personality	SMO, Random Forests, Bagging, IBk, LibSVM
(Burger et al., 2011)	Gender	char 1-5 grams, word 1-2 grams from tweets and metadata	Tweets	92%	Winnow
(Nguyen et al., 2013)	Age	Frequent unigrams	Tweets	76.70%	Logistic Regression
(Rangel and Rosso, 2013b)	Gender and Age	Word-based, punctuation marks, PoS, emoticons and affective words.	Social Media	57.13% gender, 63.50% age	SVM
(Rangel and Rosso, 2013a)	Gender and emotion	Word-based, punctuation marks, PoS, emoticons and affective words.	Facebook comments	59% gender, 59.6% joy, 32.3% anger, 36.1% disgust, 50.4% surprise, 20% sadness	J48, naïve Bayes, SVM, Bayes Nets
(Peersman et al., 2011)	Age	50,000 most distinctive unigrams	Netlog messages	71.3%	SVM
(Otterbacher, 2010)	Gender	Vocabulary richness, most common words, text complexity, usage of pronouns, oov words, metadata	IMDB comments	73.71%	Statistical Regression Model
(Kucukyilmaz et al., 2006)	Gender	Style-based features	Chat messages	81.7%	naïve Bayes
(Koppel et al., 2005)	Native lang	Function words, char n-grams and error analysis	Language learner essays	80.2%	SVM
(Wong and Dras, 2009)	Native lang	Syntactic errors, char n-grams, PoS n-grams and function words	Language learner essays	73.71%	SVM
(Tetreault and Blanchard, 2012)	Native Lang	Char n-grams, function words, PoS bigrams, spelling errors, word n-grams, perplexity scores, stanford dependencies	Language learner essays	90.1% in ICLE, 70.9%, 80.9% in TOEFL subsets, 84.6% TOEFL FULL	Logistic Regression ensemble
(Henderson et al., 2013)	Native Lang	char 1-5 grams and word 1-2 grams	Language learner essays	82.6%	Carnie, LibLinear and SRI
(Wong and Dras, 2011)	Native Lang	char 1-5 grams and word 1-2 grams and production rules	Language learner essays	81.17%	MaxEnt
(Tofighi et al., 2012)	Native Lang	lexical features, vocabulary richness, punctuation marks, word frequency and structural features	News	86.44%	SVM, C4.5, naïve Bayes
(Argamon et al., 2005)	Personality	Function words, text cohesion, assessment and appraisal measures	Student essays	58.2% neuroticism, 58% extraversion	SVM
(Oberlander and Nowson, 2006)	Personality	n-grams	Blogs	94.4% neuroticism, 90.1% extraversion, 90.4% agreableness, 92.3% conscientiousness	Naïve Bayes and SVM
(Verhoeven et al., 2013)	Personality	2000 most frequent char trigrams	Facebook data	76% extraversion, 67% neuroticism, 67% agreableness, 66% conscientiousness, 82% openness	SVM
(Poria et al., 2013a)	Personality	Word counts, number of phonemes/syllables/letters in a word, senticnet, conceptnet and emosenticnet features	Student essays	66.2% openness, 63.4% conscientiousness, 63.6% extraversion, 62.2% agreeableness, 63.7% neuroticism	SMO
(Makazhanov and Rafiei, 2013)	Political alignment	Relevant unigrams, hashtags, network-level features	Tweets	90.8%	SVM
(Kapovciute-Dzikiene et al., 2015)	Gender, Age, political alignment	Style markers, char n-grams, function words, word n-grams, lemma n-grams, PoS n-grams	Text transcripts from the Lithuanian parliament	58.7% political alignment, 74.6% gender, 44.6% age	SVM

Table 3.6: Author profiling approach summary.

3.2 Author Identification

As stated before, the main focus of this thesis is on author profiling, but given the similarities with author identification and the experiments that we performed also on that task, author identification needs to be reviewed as well, even if not that thoroughly.

The goal of author identification is to predict the author of a text given a predefined set of candidates. The methodology is similar to author profiling: the texts are processed and converted into multidimensional vectors of features that characterize the writing style of the authors. These vectors are then used to predict the authors of unseen instances.

As already in the case of author profiling, we first review in this section the used data sources and then proceed on analyzing the approaches.

3.2.1 Datasets

As in the field of author profiling, many text types have been explored in author identification. The characteristics of each kind of text have a major influence on the chosen feature set. Different kinds of features are effective in different scenarios, e.g., in chat texts, the usage of emoticons can be very effective, while in literary texts, this feature is useless.

In this section, we present the different text types that have been used in author identification.

3.2.1.1 Historic and Literary Texts

The original author identification studies were implemented to determine the authorship of disputed historic and literary texts. Some of the popular data resources are the works of Shakespeare and "The Federalist Papers".

The Federalist Papers

The Federalist Papers are a collection of essays by Alexander Hamilton, John Jay and James Madison published anonymously in 1787 to persuade the citizens of the state of New York to ratify the Constitution. Twelve of the papers were claimed to be written by Hamilton initially, but after retiring from the presidency, Madison claimed that he was the true author.

Shakespeare Authorship Studies

There have been claims that Shakespeare did not write some of his best plays. Some studies focus on the Shakespeare plays that have disputed authorship. Matthews and Merriam (1993) use a dataset of undisputed texts of Shakespeare and Fletcher as training data, and the disputed plays as test instances.

Aljumily (2015) uses a corpus that consists of 42 digital electronic copies of undisputed texts that belong to Sir Francis Bacon, William Shakespeare, John Fletcher, Christopher Marlowe, and Thomas Kyd as well as nine disputed works of Shakespeare.

Other Literary Data

Other literary texts have also been used to perform author identification. Thus, Gamon (2004) selects texts from Anne, Charlotte, and Emily Brontë. The reason behind this choice, is to perform author identification without letting other factors such as gender, education level and historic style influence the process.

Kevselj et al. (2003) use three different datasets (in English, Greek and Chinese). The English dataset considers the writings of Emily Brontë, William Shakespeare (both considered by previously described approaches), Lewis Carroll, Edgar Rice Burroughs, John Cleland, Charles Dickens, H. Ryder Haggard and Washington Irving. The Greek data is composed of news articles. The Chinese corpus includes writings from eight of the most popular modern Chinese martial art novelists.

Arabic literary texts have also been used. Altheneyan and Menai (2014) use a dataset that consists of 30 Arabic books written by 10 different authors. Each book is partitioned into chunks such that each author has 60 book chunks assigned to them.

3.2.1.2 News Articles

News articles have equally been drawn upon. In (Houvardas and Stamatatos, 2006), the authors use a subset of a news article corpus, namely, The Reuters Corpus Volume 1 (which was introduced in Section 3.1). Their training corpus consists of 2,500 texts, 50 per author. The test set contains the same amount of texts and authors, non-overlapping with the training set. All the text samples are on similar topics, which makes the problem harder to solve.

In (Zhao and Zobel, 2005), the authors use data extracted from the "TREC" corpus (Harman, 1994). This corpus contains different newswire articles that cover different writing styles and different information domains. Some of the

sources are the Wall Street Journal, the San Jose Mercury News, articles from the Computer Select disks, and the AP Newswire.

In (Raghavan et al., 2010), the authors use four datasets composed of news articles about football, business, travel and cricket.

Stamatatos et al. (2001) use texts downloaded from a modern Greek weekly newspaper called "The Tribune". Ten authors from the newspaper are selected, and thirty texts per author are included in the dataset.

3.2.1.3 Student Essays

Student essays have also been considered in the author identification literature. Van Halteren (2004) uses a dataset composed of written texts produced by first and fourth year students of Dutch philology. It contains 72 Dutch texts by 8 authors that write about pre-selected topics.

Another author attribution study that uses student essays is (Luyckx and Daelemans, 2008a). In this case, the authors use the "Personae corpus" (Luyckx and Daelemans, 2008b), which consists of 145 student essays, each approximately 1,400 words long, about a documentary on artificial life, keeping the demographic traits of the authors relatively constant. The essays contain factual descriptions of the documentary as well as the opinion of the students about it.

3.2.1.4 Email Data

Email messages have also been used in author identification, for instance by (De Vel et al., 2001), who compile a collection of 156 emails from three authors, with each author contributing with emails on three pre-established topics: movies, food and travel. Even though the corpus is small, the authors considered it sufficient for the results of their study to be significant.

3.2.1.5 Twitter Data

The work presented in (Ikeda et al., 2013), uses a dataset composed of the most recent 200 tweets of 14,000 Twitter user profiles.

Another approach that experiments with Twitter data is (Schwartz et al., 2013). A Twitter corpus composed of 500 million tweets is used. Non-English tweets, the tweets marked as retweets and tweets that contain less than three words are removed.

3.2.1.6 Chat Texts

Amuchi et al. (2012) collect 341 conversations from a chat-room based in the USA. The interventions of ten participants where the same topics are discussed,

compose the dataset.

3.2.1.7 Blog Posts

Pearl and Steyvers (2012) use blog posts. The authors use a subset of the "Spinn3r Personal Story Dataset" (Burton et al., 2009). This subset is composed of approximately 28,500 blog posts from 2,194 different authors, and contains 5.3 million words. The dataset includes authors who have between ten and twenty posts. The average length of a blog post from the final corpus is 404 words.

3.2.1.8 Forum Entries

In (Abbasi and Chen, 2005), the authors apply authorship analysis techniques to extremist-group forum messages. The authors collect English and Arabic posts extracted from specific sources: a USA-based online forum belonging to the White Knights (a chapter of the Ku Klux Klan) and forum messages associated with the Palestinian Al-Aqsa Martyrs group, which featured strong anti-American messages. In both cases, 20 messages from 20 different authors are selected.

3.2.2 Approaches to Author Identification

This section describes the most relevant works in the field of author identification grouping them with respect to the kind of data that is used.

Using Literary Texts

An example of the analysis of the authorship of Shakespeare's disputed plays is (Matthews and Merriam, 1993). The authors use neural networks, more specifically, the multilayer perceptron arrangement, and a dataset of undisputed texts of Shakespeare and Fletcher. The input of the neural network is a collection of vectors that represent word ratios. These ratios are computed using 1,000 word samples extracted from undisputed books of each author. The conclusions that the authors draw from the experiments are that "The Double Falsehood" and "The London Prodigal" have strong Fletcherian characteristics, "Henry VIII" has strong Shakespearian traits, and the characteristics of "The Two Noble Kinsmen" suggest collaboration between both authors.

Further analysis on the authorship of Shakespeare's plays is shown in (Aljumily, 2015). In this case, clustering techniques are used to verify the hypothesis that Shakespeare did write all of his disputed plays. To characterize the style, function word frequency, word n-grams and character n-grams are extracted. A dimensionality reduction process is implemented to make the problem feasible.

Hierarchical and non-hierarchical clustering techniques are used to analyze the authorship problem at hand. The authors conclude that according to empirical evidence, the hypothesis that Shakespeare is the author of the disputed plays traditionally attributed to him is false. No hypothesis on the specific authors who wrote or collaborated with Shakespeare are presented. Matthews and Merriam (1993) and Aljumily (2015) draw similar conclusions, which indicates that either Shakespeare changed his style radically for some of his works, or that some of the plays traditionally attributed to Shakespeare were not written by him.

Gamon (2004) uses deep linguistic features to differentiate between the writings of the Brontë sisters. The set of extracted features is composed of: sentence length, number of noun/adjectival/adverbial phrases and subordinate clauses per document, function word frequencies, part-of-speech trigrams, syntactic information (e.g., number and person features on nouns and pronouns, tense and aspectual features on verbs, etc.), and n-gram frequencies. The feature vectors have high dimensionality, which triggers the need for a feature selection process. As classifier, SVM is used. The presented experiments classify each author against the rest of the authors, so the experiments are binary classification problems that determine whether a text is written by a certain author or not. High values of accuracy are achieved, obtaining more than 96% in each case.

The authors of (Kevselj et al., 2003) use character-level n-grams as input for the algorithm of profile dissimilarity described in (Bennett, 1976) to determine the authors of the considered texts. To test the performance of their system, they test their approach using three datasets: an English literary dataset (that contains the works of several authors that were previously considered, such as William Shakespeare and Emily Brontë), a Greek news dataset and a Chinese literary corpus. The reported accuracies depend on the number and size of the n-grams, achieving in some cases 100% of accuracy on the English data, 95% and 97% on the Greek data and 89% on the Chinese corpus.

Using News Articles

Altheneyan and Menai (2014) use simple Naïve Bayes (NB), multinomial Naïve Bayes (MNB), multi-variant Bernoulli Naïve Bayes (MBNB), and multi-variant Poisson Naïve Bayes (MPNB) to classify Arabic literary texts by their author. The study analyzes the complex linguistic structure and challenges of the Arabic language and presents a set of 408 features that consist of the 200 most frequent words and stylistic features such as: percentage of blank lines, average sentence length, frequency of punctuation marks, percentage of function words, usage of specific function words, etc. The best result is produced by the MBNB classifier, that achieves 97.40% accuracy.

In (Houvardas and Stamatatos, 2006), the authors use a subset of The Reuters Corpus Volume 1, which consists of texts on the same topic by 50 different authors. The feature set is composed of the dominant character n-grams. A feature selection method is applied to reduce the dimensionality of the feature vectors. SVM is used to classify. The best reported accuracy predicts correctly the author of a text 74.04% of the times.

Probabilistic context free grammars (PCFGs) have been used to identify the author of texts in (Raghavan et al., 2010) (PCFGs have also been used in author profiling, as mentioned before, in (Sarawgi et al., 2011)). Five different datasets are used, four of which are composed of news articles and the fifth one of poems. The authors build a PCFG for each author, using all the documents that belong to that author in the training set. For each test document, a PCFG is built and compared with the PCFG of each author. The author whose PCFG obtains the highest similarity score is the chosen one. The performance of the PCFG similarity approach is compared with bag-of-words classification, n-gram language models and an ensemble of both baselines. The approach outperforms every baseline and the ensemble achieves accuracies of more than 90% in each case, proving that both syntactic and lexical information are useful for effectively capturing the authors' writing style.

The goal of (Stamatatos et al., 2001) is to perform author identification without relying on lexical features. To do so, three levels of features are presented: token-level (e.g., punctuation marks, number of words, number of sentences, etc.), phrase-level (e.g., percentage of noun/verb/adverbial phrases, number of words included in the noun/verb/adverbial phrases, etc.), and analysis-level (e.g., detected keywords, morphological descriptions of the words, etc.). The classification of the feature vectors is performed using discriminant analysis. A lexical approach, which uses the fifty most frequent words of the text, is implemented as baseline. The proposed approach achieves 81% of accuracy, outperforming the lexical baseline. The combination of both sets of features outperforms both of them separately, achieving 87% accuracy.

In (Zhao and Zobel, 2005), the authors present three experiments on newswire articles: classification between texts from two authors, classification between texts from five authors, and author verification (which determines whether a text is written by a given author or not). Five different classification techniques are used: Naïve Bayes, Bayes Nets, two variations of the Nearest Neighbors algorithm and Decision Trees. The authors use the frequencies of 365 function words as features. The conclusion is that the most effective method, with the best results in the majority of experiments, is the Bayesian Net classifier, and that function words are effective features to distinguish between authors.

In (Sapkota et al., 2013), the authors extract a set of features and a set of meta-features by using unsupervised techniques on the initial set of features. The

first-level features are composed of stylistic (e.g., number of sentences, tokens per sentence, usage of quotations, number of alphabetic characters, etc.), syntactic (top 1,000 part-of-speech uni/bi/trigrams and top 1,000 grammatical relations extracted with a dependency parser), semantic (top 1,000 words) and perplexity (perplexity values from character 4-grams) features. Three datasets are used: forum entries, news from topics related to business, travel, football, cricket, poems, and a subset of The Reuters Corpus Volume 1. SVM is used as classifier. The best accuracy reported on the forum data is 79% , 92.75% for the news about football, 86.66% on business, 86.8% on travel, 96.20% on cricket, 78.29% on poetry and 84.20% on the Reuters' subset.

Using Student Essays

Van Halteren (2004) uses lexical and syntactic features to predict the author of student essays. The lexical features consist of frequencies of the most frequent tokens. The considered syntactic features are constituent n-grams. Two experiments are presented: binary classification, predicting whether a text is written by a given author or not, and an 8-class classification problem that predicts the specific author of a text. Combining both sets of features and using a customized authorship scoring formula, the system predicts correctly 99.4% of the cases in the first experiment and 97% in the second.

Luyckx and Daelemans (2008a) also use student essays. They extract relevant word/part-of-speech n-grams (using the chi squared metric to select them), the most predictive function words, vocabulary richness and readability metrics. SVM is used to classify. The effect of introducing more authors into the classification process using the same feature set is studied. With two authors, the accuracy is 96.90%. Introducing extra authors makes the accuracy decrease, with still competitive accuracies of 88%, 82% and 76% for the 5, 10 and 20 author cases. A significant fall in accuracy comes when more than 50 authors are introduced, dropping to 34%.

Using Email Data

Calix et al. (2008) present a system that uses 55 stylistic features (e.g., number of words, ampersands, asterisks, semi-colons, tildes, whitespaces, number of times "anyhow" appears, etc.) to identify the authors of emails. The system is offered as a software that any non-technical user can use by introducing email data and performing the evaluation. To prove its effectiveness, twelve different authors provided ten emails on different subjects and their demographic information as well. The system is able to classify correctly every instance of this small dataset.

The K-Nearest Neighbors algorithm is used for classification.

De Vel et al. (2001) extract stylistic and structural features from emails. The stylistic features consist of characteristics such as vocabulary richness, usage of tab spaces, punctuation, upper-cased characters, and function word frequency distribution. Structural features refer to email document structural attributes such as usage of greeting/farewell acknowledgements, number of attachments, usage of signature text, etc. The experiments that are presented are all binary classifications, such that the system predicts whether a text is written by a specific author or not. Three different authors are considered. The first experiment computes the performance of the system on the full set of emails using SVM for classification. In this case, the system achieves 77.6%, 90.5% repectively 91.6% of accuracy (when distinguishing between authors 1, 2, 3 and the rest respectively). The second experiment trains the system with emails about movies and tests on the remaining topics. The results of this experiment show that the system successfully identifies two of the authors for both test topics (with more than 87% of accuracy in each case), but fails to recognize the third one, only obtaining 28.6% and 50% accuracy. It is argued that the low performance is caused by the small number of instances for the third author.

Both approaches use data compiled specifically for each work. The amount of data that is used for both approaches is quite small compared with publicly available resources such as the Enron corpus (previously used in author profiling approaches such as (Cheng et al., 2009, 2011)), which contains a large amount of texts of the same genre. The Enron corpus has also been used in author identification experiments in (Allison and Guthrie, 2008) and (Khan, 2012). Allison and Guthrie (2008) use several groups of features, namely bag-of-words, bag-of-bigrams, bag-of-trigrams, bag-of-stemmed-words and syntactic re-rule frequency features. The best result is 87.05% accuracy, obtained using the bag-of-stemmed-words features and a hierarchical probabilistic classifier (Madsen et al., 2005). Along the same lines Khan (2012) creates a bag-of-words and bag-of-bigrams feature set. This feature set is applied to the Enron email corpus and Naïve Bayes is used for classification. 90% of the mails are used for training and 10% for testing. The approach slightly outperforms (Allison and Guthrie, 2008), achieving 87.50% accuracy.

Using Twitter Data

The work presented in (Ikeda et al., 2013) describes a system that aims to identify the author of a tweet effectively. The "Source Code Authorship Profile" (SCAP) methodology is used. This methodology proceeds as follows: divides the corpus into train and test documents; concatenates for each author all the

documents into a single document; calculates the top n-grams for the combined document and calculates the "Simplified Profile Intersection" (SPI), an effective distance metric for the evaluation of the authorship of computer programs' source code (Frantzeskou et al., 2007). Each test document is assigned to the profile with the largest SPI similarity. Accuracies of over 70% using 3, 4, 5 and 6 character n-grams are reported.

Schwartz et al. (2013) also use Twitter data. The selected features are character n-grams (more specifically, 4-grams) and word n-grams (2-5grams). SVM is the chosen classifier. The authors perform author identification using 10-fold cross validation. Different configurations of training set size and number of authors are tested. In the first experiment, 50 authors are considered; 50 to 1,000 tweets form the training set. The best accuracy is obtained using the highest number of tweets: 69.7%. In the second experiment, it is shown how the accuracy of the system declines as more candidate authors are introduced (still outperforming the random classification baseline). Flexible patterns are introduced, which are defined as a generalization of word n-grams that are composed of high frequency words and content words. Repeating the first experiment (in which the training set size was variable and a fixed number of authors was considered) and introducing the flexible pattern features, the system performs better, with 2.9% of improvement. A comparison between this method and competitive approaches is provided, where the introduced method outperforms all the competitors.

Using Chat Messages

The analysis of chat messages can be applied to many real-world applications. As a result, chat messages have been used in both author profiling (Kucukyilmaz et al., 2006) and author identification (Amuchi et al., 2012). Amuchi et al. (2012) analyze the growing phenomenon of online grooming[20]. To contribute to the cause of identifying potential predators, the authors apply author identification techniques to assess whether specific users of chat sites can be identified. Different classification methods are used: Naïve Bayes, SVM, Bayesian Regression, simple Markov chains, and Chi Square Coefficient. Five feature groups are taken into account: lexical (e.g., words per sentence, word length distribution, characters per sentence, vocabulary richness, etc.), syntactic (punctuation marks, function words and their usage patterns), n-grams, structural (e.g., font size, use of audiovisuals, signatures, etc.), and content-specific (i.e., words that are very specific to a certain topic domain). Applied to the whole feature set, an SVM classifier identifies correctly the author in 85.80% of the times.

[20] A possible definition of online grooming could be the following: Befriending and establishing an emotional connection with a child, and sometimes the family, for child sexual abuse.

Using Blog Posts

Pearl and Steyvers (2012) focus on authorship verification. The goal of the study is to implement a system that, given a target document and a set of documents from a known author, decides whether the document is written by the same author or not. To do so, the authors extract a whiteprint characterization of the documents that consists of the extraction of stylometric features. The stylometric features are composed of the following feature groups: character frequencies, usage of punctuation marks, part-of-speech frequencies, lexical diversity, first person pronoun usage, average sentence/word length and total number of words. Two practical demonstrations are presented. The first one uses blog posts and the second data gathered from writers who specifically tried to imitate an existing author. In the first experiment, the system achieves 89% accuracy using the full feature set; the second one obtains a perfect score of 100%. The chosen classification method is a sparse multinomial logistic regression algorithm. Even though blog posts have been used widely in author profiling works (e.g., (Schler et al., 2006; Goswami et al., 2009; Nguyen et al., 2011; Argamon et al., 2009)), in author identification, the usage of this type of texts is rare.

Using Forum Entries

In (Abbasi and Chen, 2005), the authors apply authorship analysis techniques to forum messages written in English and Arabic by members of extremist groups. Arabic is a language that is not often considered in author identification experiments, but was also used in (Altheneyan and Menai, 2014). SVM and C4.5 are the chosen classifiers. A set of lexical (e.g., letter frequency, vocabulary richness, elongation, special character usage, etc.), syntactic (punctuation, function words and word root analysis), structural (text structure and meta-features such as font color or size), and content-specific features are used. The features differ in some details from one language to the other due to the specific characteristics of the Arabic language. SVM is the classifier that performs best, with accuracies of 97% on the English data and 94.83% on the Arabic dataset. A comparison of feature usage for both datasets is presented, providing insight on the writing style of both extremist groups.

3.2.3 Shared Tasks and Related Topics

Shared tasks about author identification are organized on a regular basis. In these tasks, the users are asked to identify the author of a text from a pool of possible authors. See (Argamon and Juola, 2011; Juola, 2012; Juola and Stamatatos, 2013; Stamatatos et al., 2014, 2015), for overview papers that describe the data and

presented approaches. Related shared tasks are plagiarism detection (see (Eiselt and Rosso, 2009; Potthast et al., 2011, 2013)) and sexual predator identification (see (Inches and Crestani, 2012)).

Other relevant shared tasks that are in line with author profiling also need to be mentioned: racism detection (Tulkens et al., 2016), hate speech identification (Haji Mohammad et al., 2016), automatic detection of cyberbullying (Van Hee et al., 2015a,b), and a study on the characteristics of pedophile conversations (Gupta et al., 2012).

Another related task is deception detection. In this case, words are analyzed to determine whether the speaker is lying or telling the truth. See e.g., (Newman et al., 2003; Feng et al., 2012; Adams, 1996; Vrij, 2008).

3.2.4 Summary

To summarize the state-of-the art of author identification, again two tables are provided. Table 3.7 shows the publicly available resources that are available for author identification.

ID	Resource Name	Size	Language	Labeling	Source	Reference
EC	Enron Corpus	$>$ 500K texts	EN	Author	Emails	(Shetty and Adibi, 2004)
BNC	The British National Corpus	4049 texts	EN	Gender, Age Author Geographic origin	Formal Texts	(Clear, 1993)
P11AI	PAN11 author identification task corpus	9337 texts	EN	Author	Emails	(Argamon and Juola, 2011)
P12AI	PAN12 author identification task corpus	$<$ 300 texts	EN	Author	Fiction Texts	(Juola, 2012)
P13AI	PAN13 author identification task corpus	$<$ 400 texts	EN, ES, GR	Author	Academic Texts, News Articles, Short Fiction	(Juola and Stamatatos, 2013)
P14AI	PAN14 author identification task corpus	$>$ 4000 texts	EN, ES, GR, NL	Author	Essays, Reviews, Novels, Articles	(Stamatatos et al., 2014)
P15AI	PAN 15 author identification task corpus	$>$ 4000 texts	EN, ES, GR, NL	Author	Not Specified	(Stamatatos et al., 2015)

Table 3.7: Publicly available resources for author identification.

Table 3.8 provides a summary of the different approaches applied to author identification.

Reference	Features	Corpus	Accuracy	Classifier
(Matthews and Merriam, 1993)	1.000 most used words	Literary texts	$> 90\%$	Multilayer Perceptron
(Aljumily, 2015)	Function words word/char n-grams	Literary texts	-	Hierarchical Clustering
(Gamon, 2004)	Function words syntactic features word frequencies n-gram frequencies, noun/adjective/adverbial phrase frequency.	Literary texts	$>96\%$	SVM
(Kevselj et al., 2003)	Char n-grams	Literary texts News articles	$> 90\%$	Profile Dissimilarity
(Altheneyan and Menai, 2014)	200 most frequent words, punctuation frequency function words, sentence length	Arabic literary texts	97.40%	Naïve Bayes
(Houvardas and Stamatatos, 2006)	Character n-grams	News articles	74.04%	SVM
(Raghavan et al., 2010)	PCFG grammars	News articles, poems	$> 90\%$	PCFG similarity
(Stamatatos et al., 2001)	Token-level features, phrase-level features, keyword frequency, morphological information	Greek News articles	81%	Discriminant analysis based on mahalanobis distance
(Zhao and Zobel, 2005)	Function words	News articles	$> 80\%$	Naïve Bayes, Bayes nets, Knn, Decision Trees
(Sapkota et al., 2013)	Simple stylistic features, syntactic features, semantic characteristics, meta-features	Forum entries, football articles, reuters articles	79% (forum entries) 92.75% (football article) 84.20% (Reuters)	SVM
(Van Halteren, 2004)	Lexical and syntactic features	Student essays	$> 97\%$	Custom authorship score formula
(Luyckx and Daelemans, 2008a)	Word/PoS n-grams, function words, vocabulary richness readibility metrics	Student essays	$> 80\%$	SVM
(Calix et al., 2008)	Stylistic features	Emails	100%	Knn
(De Vel et al., 2001)	Stylistic and structural features	Emails	$> 77\%$	SVM
(Ikeda et al., 2013)	Top n-grams	Tweets	$> 70\%$	SPI Similarity
(Schwartz et al., 2013)	Char/word n-grams	Tweets	69.7%	SVM
(Amuchi et al., 2012)	Word frequency, syntactic features specific keywords	Chat messages	85.80%	Naïve Bayes SVM, Bayessian Regression, Markov Chains Chi squared
(Pearl and Steyvers, 2012)	Character frequencies, PoS, lexical diversity, first person pronoun usage, sentence/word length total number of words	Blog posts	$>89\%$	Multinomial Logistic Regression
(Abbasi and Chen, 2005)	Lexical, syntactic, structural and content-specific features	Forum entries	$>94\%$	SVM, C4.5

Table 3.8: Author identification approach summary.

Chapter 4

FEATURE ENGINEERING

In Chapter 2, we discussed the relevance of feature engineering in machine learning problems. In this chapter, our application of feature engineering to author profiling and identification is presented.

First, to understand the basics of feature engineering in the context of author profiling/identification, an example is provided. In this example, different types of features that could be used to solve the task are introduced and the different aspects of the text that can be analyzed and specific feature values are shown. Then, the resources that were created during the development of the thesis are introduced. After that, we present our feature set, motivate the choice of every feature, evaluate its relevance and compare it to alternative, commonly used feature sets. Finally, we draw some conclusions.

4.1 Example of feature engineering

To illustrate how author profiling/identification methods and the process of feature engineering work, two text fragments written by two different authors are presented, analyzed and discussed at different levels. Both texts are fragments of novels written by British authors in a similar time period.

The first fragment is written by Charles Dickens in "A tale of two cities":

> The transition to the sport of window-breaking, and thence to the plundering of public-houses, was easy and natural. At last, after several hours, when sundry summer-houses had been pulled down, and some area-railings had been torn up, to arm the more belligerent spirits, a rumour got about that the Guards were coming. Before this rumour, the crowd gradually melted away, and perhaps the Guards came, and perhaps they never came, and this was the usual progress of a mob.

The second one is written by Arthur Conan Doyle in "The adventures of Sherlock Holmes":

> Sherlock Holmes was wrong in his conjecture, however, for there came a step in the passage and a tapping at the door. He stretched out his long arm to turn the lamp away from himself and towards the vacant chair upon which a newcomer must sit. "Come in!" said he. The man who entered was young, some two-and-twenty at the outside, well-groomed and trimly clad, with something of refinement and delicacy in his bearing.

We can analyze two main aspects of these texts: content and structure. Both options have been explored in the state of the art, but, as became clear in Chapter 3, content-based approaches are more common.

If we decide to analyze the content of the text, a classic strategy would be to use the most relevant words in the corpus to classify the texts. There are many different strategies to decide which words are the most relevant: e.g., the most frequent words, the words with a higher tf-idf coefficient, the words that have higher Information Gain, etc. If we decide to use the ten most frequent words in the corpus (in this case composed only of these two texts), we have 10-dimensional feature vectors that contain the frequencies of the following ten words: *the*, *and*, *a*, *to*, *of*, *was*, *his*, *in*, *had*, *at*, for each instance.

The resulting feature vectors are the following:

> instance1: [the: 7/79, and: 6/79, a: 2/79, to: 3/79, of: 3/79, was: 2/79, his: 2/79, in: 0/79, had: 2/79, at: 1/79]
>
> instance2: [the: 5/74, and: 4/74, a: 3/74, to: 1/74, of: 1/74, was: 2/74, his: 3/74, in: 3/74, had: 2/74, at: 2/74]

The vectors would be computed for each instance of the corpus, and the vectors with their correct labels would be passed to a classifier. The classifier would learn from the provided training data and use the extracted knowledge to predict the author of unseen instances. This type of approach is often implemented using large amounts of words (in some cases, the frequencies of more than 1,000 words are used, such that the feature vector of each instance has a very high dimensionality). This kind of approach is usually referred to as "bag of words" because the words are not ordered with respect to their relevance or other criteria.

Character or token n-grams are other options that have proven in the past to be very effective. In the case of character n-grams, the most relevant combinations of n consecutive characters are used for classification. In the case of token n-grams, the most relevant sequences of n consecutive tokens are considered. The

choice of both the number of n-grams to select and the relevance criteria are the key aspects of this sort of approaches. Thousands of n-grams are usually selected using criteria such as frequency, tf-idf coefficient or Information Gain.

If we think about the different levels of a text that can be analyzed, and start with the smaller units, characters are the first level. At the character level, we can analyze the usage of certain specific characters, the most stylistically relevant being punctuation marks. The usage of commas, in particular, can be very stylistic: commas change the intonation of the reader of the text, can be used to denote a simple pause when reading, to imitate speech patterns, etc. They can even be omitted to speed up the pace of a sentence. Measuring the frequency and pattern of the usage of commas can be a very effective way to characterize the writing style of an author. Analyzing comma usage, we can see in our example that even if both texts are similar in length (79 and 74 words respectively), Dickens uses 11 commas and Conan-Doyle only 4.

Other punctuation marks are also very interesting: a frequent usage of periods can be related to shorter sentences and exclamation/interrogation marks can give information on the general tone of the text. Capitalization is another relevant characteristic: high amounts of capitalized letters with respect to the total number of sentences in a text can be related to a high number of proper nouns.

At the word level, we can use metrics such as the mean number of characters per word and the vocabulary richness. Measuring the mean number of characters per word, we can observe the mean word complexity, which can serve as a simple indicator on the complexity of the narrative. The value of the first measure in the first text is 5.0 and 4.64 in the second case, which are similar. Vocabulary richness computes the percentage of different words used, with respect to the total number of words in a text. The values of vocabulary richness are 0.69 respectively 0.78, indicating that Conan-Doyle uses in the presented case a higher number of unique words than Dickens.

Measuring the usage of function words is also a very powerful profiling feature (as we have seen in Chapter 3, many state-of-the-art approaches use function words in author profiling/identification). Function words are words that have little lexical meaning but that serve to express grammatical relationships with other words within a sentence. Therefore, even if they do not provide meaning, they are important to the structure of sentences. Function words are of relevance also because some of them (for instance, conjunctions and interjections) are highly stylistic. So, a focus on specific types of function words, and specific word choices can be very helpful to profile the authors. In our example texts, the usage of "thence" in the first fragment is interesting and can be seen as a stylistic choice of the author.

Sentence-level analysis can be the next step. The mean number of words per sentence is one of the metrics that first comes to mind. The first text fragment,

composed of three sentences, contains 26.33 words per sentence, and the second one, composed of 4, 18.5, which makes Dickens' sentences significantly longer.

After looking at the content of the sentences and analyzing their basic characteristics, we can analyze the syntactic structure of the sentences of the texts. The most basic morpho-syntactic characteristics that can be used are the parts of speech of the used words. A high amount of adjectives can be a result of a very descriptive text, and frequent verb usage can result in a more action-based narrative.

Syntactic trees per sentence can be derived for both texts. Using these syntactic trees, several features can be computed. Constituency-based or dependency-based parsers can be used to derive these structures. To give an example of the kind of information that can be extracted from dependency trees, the trees of two sentences (one per fragment) are provided in Figure 4.1 and Figure 4.2.

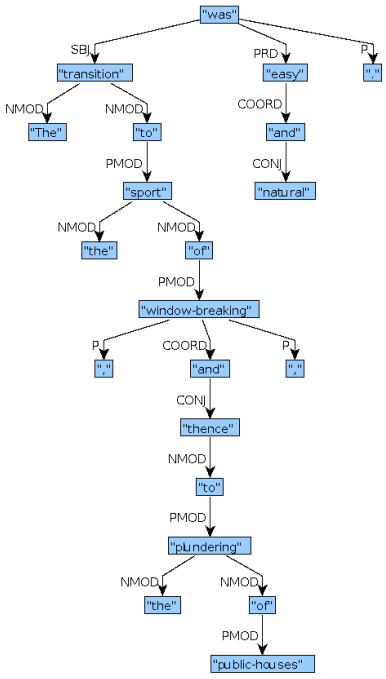


Figure 4.1: Dependency tree of a sentence by Dickens.

In these trees, each node is a word of a sentence; the arcs are syntactic dependencies. The specific dependency relation set that we use is introduced in the next section. Each dependency relation gives information about the nature of the link between two lexical items. A word can be a modifier of another word; a dependency can also indicate that a subordinate clause starts, that a word serves as an

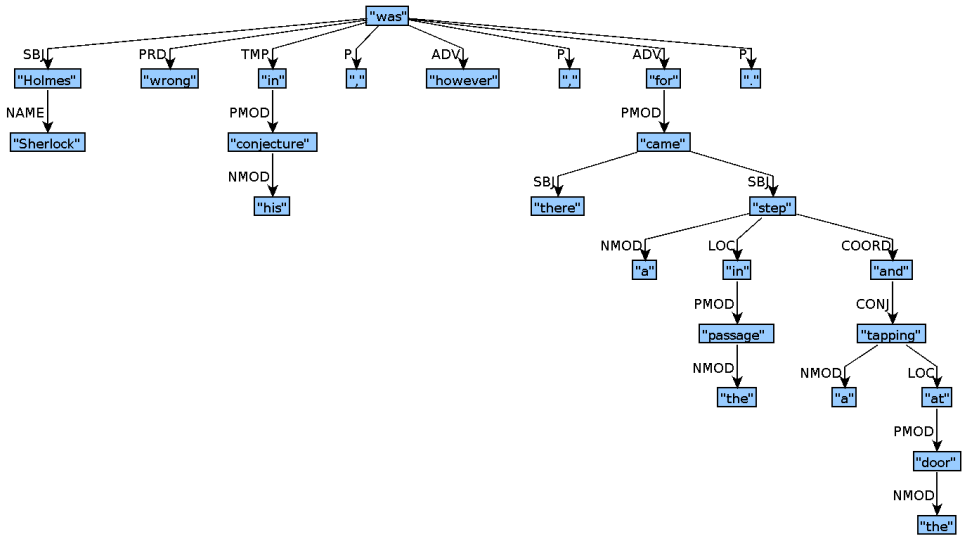


Figure 4.2: Dependency tree of a sentence by Conan-Doyle.

adverbial (e.g., *very fast*, *too weak*), or that we face a complex verbal construction (e.g., *has been*, *has taken*). The frequency of each dependency can provide very interesting stylistic information. For instance, a high number of subordinate clauses indicates a high complexity of the narrative.

To analyze syntactic structures from a classic data structure point of view can also be very helpful. Thus, the specific shape of a tree can give useful information on the complexity of the inner structure of sentences. The presented trees have very different shapes: Dickens' sentence is deep and narrow, while Conan-Doyles' is wide and not as deep. In the first case, we can see that the depth of the sentence resides in the high level of detail dedicated to one idea (a transition to violence); in the second, more ideas are presented without detailing them that much.

4.2 Resources

Before starting to explain the feature set, the resources that were created in the course of the work on this thesis need to be introduced. These resources will be used throughout the thesis.

To obtain an overview of the characteristics of these resources, a Table 4.1 is presented. As shown, many different resources have been created. The majority of them are composed of texts written in English, but texts in other languages such as French, Catalan, German, Italian and Spanish are also included. To provide

more information about the resources, the data sources per corpus are displayed in Table 4.2.

Identifier	Genre	Labels	Lang	MeanChars	MeanWords	MeanSents	#Authors	#texts
NYTimes	Opinion Blog Posts	gender	En	6191.21	1157.28	41.23	>200	1,672
EnglishDat	Opinion Columns	gender	En	2127.37	398.55	12.79	51	7,148
FrenchDat	Opinion Columns	gender	Fr	2298.63	417.56	13.24	18	4,310
CatalanDat	Opinion Columns	gender	Cat	2382.74	455.86	16.88	33	4,078
GermanDat	Opinion Columns	gender	De	3568.77	572.07	24.73	127	3,564
ItalianDat	Opinion Columns	gender	It	1680.61	300.80	11.19	43	4,265
SpanishDat	Opinion Columns	gender	Es	3568.42	695.23	23.33	101	5,794
SmallEngDat	Opinion Columns	gender author	En	4696.03	932.92	30.09	35	1,260
AuthorshipDat	Opinion Blog Posts	gender author	En	4262.15	824.75	26.80	26	5,118
LiteraryBritish	Novel Chapters	gender author	En	19997.82	4258.62	192.76	18	1,793
LiteraryAmerican	Novel Chapters	gender author	En	18137.55	3949.48	181.54	17	1,570
LiteraryMerged	Novel Chapters	gender author origin	En	19129.37	4114.30	187.52	35	3,364

Table 4.1: Resource summary.

Most of the sources crawled for the development of the resources are online newspapers, where blog posts and opinion columns are easily obtainable and the authors of the texts are not anonymous. For the literary resources, the Project Gutenberg website was used, where full books are available.

4.3 Feature Set

In this section, we present each feature that is used in our experiments. However, before getting into specific feature descriptions, we need to understand the basic flow of our approach. Figure 4.3 shows an overview of how our system works.

As we can see, the input data is raw text. Before the feature extraction process is performed, syntactic dependency and discourse trees are constructed. Then, the raw text, is fed along with both sets of trees into our feature extractor, which transforms this input into multidimensional vectors, one per instance, with each feature value as a dimension.

These vectors and the ground truth label of each instance are given as input to a machine learning algorithm, which extracts patterns from the training material and makes predictions on unseen instances. To predict, we either provide the machine learning algorithm with the feature vectors of a test set, separated from the initial training set, or perform 10-fold cross validation. The particular approach depends on the nature of the experiment in question, not all features are used in all

Identifier	Sources
NYTimes	NYTimes Opinionator blog
EnglishDat	The Sun, The Times New York Daily
FrenchDat	L'express, Le Monde
CatalanDat	El Punt Avui, Ara, Mes, Directe
GermanDat	Die Welt, Süddedeutsche Zeitung, Frankfurter Allgemeine Zeitung, Compatct, taz
ItalianDat	Corriere della Sera, Ill Messaggero, Il Post
SpanishDat	Publico, El Mundo, La Vanguardia, 20 Minutos, ABC, El Periodico
SmallEngDat	Dallas News, NYDaily News, Canberra Times, The Telegraph, The Guardian, The Independent
AuthorshipDat	The Guardian, The Independent, The Daily Mail
LiteraryBritish	Project Gutenberg
LiteraryAmerican	Project Gutenberg
LiteraryMerged	Project Gutenberg

Table 4.2: Sources used per resource.

experiments, and in some occasions, supervised learning is not the machine learning technique of choice. More detailed, experiment-specific setups are introduced in Chapter 5.

Our feature set is divided into six feature subgroups, each one analyzing the text at a different level. This specific feature set has been successfully used in different tasks and can be easily expanded.

4.3.1 Character-based features

The first analysis level of our feature set is character-based. The usage of relevant characters such as punctuation marks is computed. The values of each feature are relative to the length of the text to make these feature values length-independent (otherwise, text length variability would influence feature values). For each feature, the formula used to compute the feature value and a justification of its relevance is provided afterwards. The features that compose this group are the following:

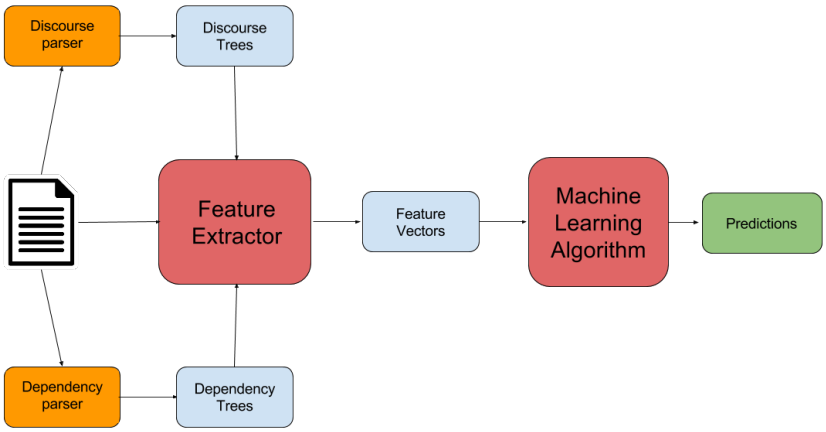


Figure 4.3: Basic data and control flow of our approach.

Upper cases

$$UpperCaseUsage = \frac{Number\_of\_upper\_cased\_chars}{Total\_chars} \tag{4.1}$$

This feature can be useful to measure the usage of proper nouns, acronyms or, in informal writing, the emphasis on certain parts of the text (e.g., using the token "HELLOOOOO" instead of the regular "hello").

Numeric digits

$$NumberUsage = \frac{Numbers\_in\_the\_text}{Total\_chars} \tag{4.2}$$

The usage of numeric digits can be a stylistic choice or a topic characteristic. Texts on certain topics (e.g., scientific articles) have a tendency of using numeric values, but in general, the usage of numbers instead of the corresponding words can be considered a stylistic choice.

Commas

$$CommaUsage = \frac{Commas\_in\_the\_text}{Total\_chars} \tag{4.3}$$

As stated before, (especially in English) commas are very much stylistic elements. That is, when the use of a comma is not grammatically required, the

decision to use a comma depends on the writer's goal. Commas can be used to emphasize a part of the sentence creating pauses both before and after the word as in the following sentence:

"For men, love is a thing formed by equal parts of lust and astonishment."

In this case, the writer wants to emphasize that the statement is about how **men** perceive love.

Commas also determine the reading rhythm of the reader and are used by writers to implement their specific pace. This feature has proven to be very useful in characterizing the writing style of a specific author/author profile.

Periods

$$PeriodUsage = \frac{Periods\_in\_the\_text}{Total\_chars} \tag{4.4}$$

It is useful to capture the usage of periods in a text in order to measure the sentence length indirectly. This feature is complemented by the next one.

Omission

$$EllipsisUsage = \frac{SuspensionPoints\_in\_the\_text}{Total\_chars} \tag{4.5}$$

This feature (encoded as suspension points) computes the usage of omissions, words that are implied by the reader without being explicitly mentioned. An omission can also be an indicator of an unfinished sentence, a pause, a signal of confusion/disapproval/hesitation or a very weighty pause. This technique is also used when partially quoting a fragment of a text. Consider some examples:

(as an omission) *She ran, but I don't know why...*
(as a quote)*...In proportion as he simplifies his life, the laws of the universe will appear less complex...*
(as a sign of hesitation) *Um... I'm not sure about that.*
(as a pause) *As to what we do next...*

So, measuring the number of suspension points (counting the number of times, the three suspension points are found) in relation to the total number of characters in the text, we measure the usage of omissions in the text.

Question marks

$$QuestionUsage = \frac{Questions\_in\_the\_text}{Total\_chars} \tag{4.6}$$

The use of question marks can indicate a dialog between characters (questions asked between characters), inner dialog or rhetorical questions. It is a valuable punctuation mark to be taken into account at this level.

Exclamation marks

$$ExclamationUsage = \frac{Exclamations\_in\_the\_text}{Total\_chars} \tag{4.7}$$

Exclamation marks are good indicators of the expressiveness of a text. The tone of the following sentences is radically different, depending on whether an exclamation mark is used or not:

Luis was here yesterday.

Luis was here yesterday!!!!

It is rather obvious that the fact that Luis was around was an extraordinary event in the second case, while in the first one, it was something mundane.

This feature is very useful to analyze the expressiveness of the writer and the general tone of the text.

Colon ratio

$$ColonUsage = \frac{Colons\_in\_the\_text}{Total\_chars} \tag{4.8}$$

Colons precede explanations, enumerations or lists. Measuring these phenomena in the text, can provide very useful insight for profiling the style of the author.

Semicolon ratio

$$SemiColonUsage = \frac{SemiColons\_in\_the\_text}{Total\_chars} \tag{4.9}$$

Semicolons separate major sentence elements. They can be used between two independent clauses, as a replacement for commas, or as a list separator. The usage of semicolons is usually motivated by long sentences, where different topics are introduced. If an author uses semicolons often, it can be due to the complexity and variety of the ideas presented in his/her sentences.

Characters between parentheses

$$CharsParenthUsage = \frac{Chars\_between\_parentheses}{Number\_of\_parentheses} \tag{4.10}$$

This feature measures the mean number of characters found between parentheses. Usually, parentheses are used to clarify aspects of the preceding sentence or lexical element. The usage of this type of clarification can be very stylistic. High values of this feature can be caused by the desire of the author that the text is entirely understood by the reader.

Parentheses usage

$$ParenthesesUsage = \frac{Parentheses\_in\_the\_text}{Total\_chars} \tag{4.11}$$

Closely related to the previous feature, this one computes the number of parentheses used in a text, which corresponds to the number of cases where the author has made important remarks about the previous sentence or lexical element in order to make it completely understood.

Hyphen ratio

$$HyphenUsage = \frac{Hyphens\_in\_the\_text}{Total\_chars} \tag{4.12}$$

The ratio of hyphens in the text measures the occurrence of a dialog in the text or the usage of hyphenated words[1]. It is another factor to be taken into account.

Quote usage

$$QuoteUsage = \frac{Quotes\_in\_the\_text}{Total\_chars} \tag{4.13}$$

To measure quote usage, all possible quotation characters are considered (‘’, “”, «»). This feature is an indicator of dialog, of direct speech, citations, examples, or irony. Consider a clear example of marking irony by quotes:

Today, the “professional” IT guy, came to fix the printer.

It’s obvious that the “IT guy” is not considered professional at all, and that the writer explicitly indicates this. This usage of quotes is very stylistic and has proved to be a very effective feature.

[1] Although the use of hyphens as a punctuation mark can theoretically be distinguished from their use in hyphenated words, in practice, this distinction cannot be made reliably.

4.3.2 Word-based features

The next feature group analyzes the text at the word level. Instead of using directly the words that occur more frequently (or the most relevant ones) for classification (as in bag-of-words approaches), we measure the characteristics and usage of certain types of words. As in the previous group, each feature value is relative to the number of words per text in order not to depend on text length.

To obtain text tokenization (the list of words used in the text), the word tokenizer from Python's Natural Language Toolkit is used.

The features that compose this feature group are the following:

Characters per word

$$CharsPerWord = \frac{Total\_chars}{Total\_words} \tag{4.14}$$

The first feature of this group is quite intuitive. Measuring the mean number of characters per word is a very good way to measure the complexity of the words that the author uses. If the mean number of characters per word is high, the author tends to use long (and possibly) complex words. This can be due to the topic that is being written about, or to the stylistic choices of the author.

Standard deviation of characters per word

$$STDCharsPerWord = \sqrt{\frac{\sum_{i=1}^{N}(CharsPerWord_i - meanCharsPerWord)^2}{N}} \tag{4.15}$$

This formula computes the standard deviation of characters per word. It measures the spread of the feature values around the mean value. It is a useful metric that allows us to see whether the variation in word length is high. For instance, if we have the following two cases of word lengths:

15, 15, 15, 14, 16
2, 7, 14, 22, 30

both of them have the same mean number of characters per word, but the standard deviation of the second case is much higher than of the first, indicating that the first uses words with similar length regularly, while the second one is much more variant.

Range of characters per word

The statistical range is defined as the difference between the largest and the smallest element of a set. In this case, we substract the shortest word length from the longest one. Combined with the previous two features, this coefficient can be a useful feature to characterize the word choices of a specific author/profile.

Vocabulary richness

$$VocabularyRichness = \frac{Different\_words}{Total\_words} \tag{4.16}$$

Vocabulary richness measures the percentage of words that are used in a text uniquely. It is a measure that has been very effective in author profiling (especially in the differentiation between male and female writings), and that is a metric on how diverse the vocabulary of an author is.

Acronym usage

$$AcronymUsage = \frac{Acronyms\_in\_the\_text}{Total\_words} \tag{4.17}$$

This feature computes the ratio of tokens in a text that are acronyms. An acronym is a word or name formed as an abbreviation of the parts of a sentence, multiple word expression, or a word (e.g., Lysergic Acid Diethylamide is widely known as LSD).

Stopword usage

$$StopwordUsage = \frac{Stopwords\_in\_the\_text}{Total\_words} \tag{4.18}$$

Stopwords are common words that usually do not contribute significant meaning to the text. As a result, stopwords are considered to be of little importance in search processes and are filtered out. On the other side, a low amount of stopwords could mean that the author of a text is using a high amount of infrequent words. This information could be very useful to profile the writing style of the author.

To implement this feature, the stopword list provided by Python's Natural Language Toolkit (NLTK) has been used.

First person pronoun usage

$$FPSingPronUsage = \frac{FPSingularPronouns\_in\_the\_text}{Total\_words} \tag{4.19}$$

$$FPPlurPronUsage = \frac{FPPluralPronouns\_in\_the\_text}{Total\_words} \tag{4.20}$$

This feature could be classified either under this or under the syntactic category. The percentage of first person pronouns (both singular and plural) is measured. In English, these pronouns are:

I, *me*, *my*, *mine* (singular)
We, *our*, *ours*, *us* (plural)

Measuring the usage of these pronouns is very useful when analyzing non-fiction texts such as opinion columns, blog posts or social media content. It reveals the tendencies of an author to write about themselves as individuals or as a group, and is thus useful in the context of age/gender identification.

Words between parentheses

$$WordsParenthUsage = \frac{Words\_between\_parentheses}{Number\_of\_parentheses} \tag{4.21}$$

Similar to the corresponding character-level feature, the mean number of words appearing between parentheses is measured.

Two and three character word usage

$$TwoCharWordUsage = \frac{TwoCharWords\_in\_the\_text}{Total\_words} \tag{4.22}$$

$$ThreeCharWordUsage = \frac{ThreeCharWords\_in\_the\_text}{Total\_words} \tag{4.23}$$

This feature is divided into two: we measure the percentage of words that are two and three characters long. It is another measure to see how complex the words used by the authors are.

4.3.3 Sentence-based features

This feature group is fairly simple: we compute the mean number of words per sentence and its standard deviation and range. It is a way to characterize, in a shallow way, the basic structure of the sentences of a text.

The sentence splitting is performed using the "punckt" sentence splitter from Python's NLTK.

The features in this group are the following:

Words per sentence

$$WordsPerSentence = \frac{Total\_words}{Total\_sentences} \tag{4.24}$$

Measuring the mean number of words per sentence in a text is another way to characterize sentence complexity. To use long sentences implies complex writing, which can be related to characteristics of the author: their particular writing style (if we are analyzing, for example, different writers from a variety of literary periods), gender, age and personality, among other possibilities. It is a simple but very useful feature, which is complemented by computing its standard deviation and range.

Words per sentence standard deviation

$$STDWordsPerSent = \sqrt{\frac{\sum_{i=1}^{N}(WordsPerSent_i - meanWordsPerSent)^2}{N}} \tag{4.25}$$

As in the case of the "characters-per-word-STD" feature, we measure the variability of words per sentence by computing the standard deviation. It is a metric that indicates whether an author constructs his/her sentences in a uniform way, or whether they vary greatly in size from one sentence to another.

Range of words per sentence

This feature computes the statistical range of words per sentence. It is computed by substracting the length of the shortest sentence from the length of the longest one.

4.3.4 Dictionary-based features

The features of this category are computed using dictionaries that contain words of specific types. It is the most content- and language-dependent feature group. The dictionaries are either publicly available resources or compiled for the development of the thesis. The dictionary-based features and the resources used to compute them, are the following:

Discourse marker usage

$$DiscourseMarkerUsage = \frac{Discourse\_markers\_in\_the\_text}{Total\_words} \tag{4.26}$$

Discourse markers are words that mark the flow and structure of discourse. They can be described as the "glue" that binds together a piece of writing, ensuring its coherence. Some widely used discourse markers are for instance: *in conclusion*, *to begin with*, *firstly*, *moreover*, *on the other hand* and *however*. The list of discourse markers that we consider was extracted from:

`https://aliciateacher2.wordpress.com/grammar/discourse-markers/`.

This feature is complemented by the discourse features, which explore in a deeper way the specific discourse structure of the texts (see Section 4.3.6 below).

Interjection usage

$$InterjectionUsage = \frac{Interjections\_in\_the\_text}{Total\_words} \tag{4.27}$$

An interjection is defined as an expression that is inserted into an utterance without grammatical connection to it; an interjection tends to express emotions. Some sample interjections are, for instance: *oops*, *golly*, and *gosh*.This feature is particularly useful when differentiating between writers of different ages: young writers have a tendency to write in a more spontaneous way, shown in part by their tendency to use interjections more frequently than adult writers. The interjection list that has been used was extracted from

`http://www.yourdictionary.com/index.php/pdf/articles/156.listof-interjections.pdf`.

Polar words

$$PositiveWordUsage = \frac{PositiveWords\_in\_the\_text}{Total\_words} \tag{4.28}$$

$$NegativeWordUsage = \frac{NegativeWords\_in\_the\_text}{Total\_words} \tag{4.29}$$

These features measure the usage of positive and negative sentiment words. To determine which words are positive and which ones are negative, a sentiment analysis lexicon is used. The lexicon contains a list of words that belong to each

category. This resource is publicly available[2], and is described in (Hu and Liu, 2004).

The tendency of using positive and negative polarity words is in some cases directly related to the writers' personal situation and personality. There are stereotypical behaviours that can be analyzed by measuring the usage of this type of words. A clear example of that is gender identification, where polarity word features have proven to be very effective, showing that female writers tend to use more polar words than men.

Sensation word usage

$$SensationWordUsage = \frac{SensationWords\_in\_the\_text}{Total\_words} \tag{4.30}$$

$$AfraidWordUsage = \frac{AfraidWords\_in\_the\_text}{Total\_words} \tag{4.31}$$

$$AmusedWordUsage = \frac{AmusedWords\_in\_the\_text}{Total\_words} \tag{4.32}$$

$$AngryWordUsage = \frac{AngryWords\_in\_the\_text}{Total\_words} \tag{4.33}$$

$$AnnoyedWordUsage = \frac{AnnoyedWords\_in\_the\_text}{Total\_words} \tag{4.34}$$

$$DontCareWordUsage = \frac{DontCareWors\_in\_the\_text}{Total\_words} \tag{4.35}$$

$$HappyWordUsage = \frac{HappyWords\_in\_the\_text}{Total\_words} \tag{4.36}$$

$$InspiredWordUsage = \frac{InspiredWords\_in\_the\_text}{Total\_words} \tag{4.37}$$

$$SadWordUsage = \frac{SadWords\_in\_the\_text}{Total\_words} \tag{4.38}$$

$$AfraidWordRatio = \frac{AfraidWords\_in\_the\_text}{Total\_sensationWords} \tag{4.39}$$

[2]The lexicon can be downloaded from the following website: `https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon`

$$AmusedWordRatio = \frac{AmusedWords\_in\_the\_text}{Total\_sensationWords} \tag{4.40}$$

$$AngryWordRatio = \frac{AngryWords\_in\_the\_text}{Total\_sensationWords} \tag{4.41}$$

$$AnnoyedWordRatio = \frac{AnnoyedWords\_in\_the\_text}{Total\_sensationWords} \tag{4.42}$$

$$DontCareWordRatio = \frac{DontCareWors\_in\_the\_text}{Total\_sensationWords} \tag{4.43}$$

$$HappyWordRatio = \frac{HappyWords\_in\_the\_text}{Total\_sensationWords} \tag{4.44}$$

$$InspiredWordRatio = \frac{InspiredWords\_in\_the\_text}{Total\_sensationWords} \tag{4.45}$$

$$SadWordRatio = \frac{SadWords\_in\_the\_text}{Total\_sensationWords} \tag{4.46}$$

The above features complement the previous ones. Instead of measuring the usage of positive and negative words, they measure the usage of words that evoke certain sensations: fear, amusement, anger, annoyance, indifference, happiness, being inspired and sadness. For each sensation, we measure the percentage of words in a text that evoke this sensation as well as the percentage of the sensation words that belong to each category. The percentage of sensation words with respect to the total number of words is also computed.

To extract these features, a publicly available resource called "Depeche Mood" is used, which provides dictionaries for each one of the mentioned sensations. It can be downloaded from the following url:

`https://github.com/marcoguerini/DepecheMood/releases`

The reference paper of this resource is (Staiano and Guerini, 2014).

Curse words

$$CurseWordUsage = \frac{CurseWords\_in\_the\_text}{Total\_words} \tag{4.47}$$

Measuring the tendency of an author to use curse words can be very useful, especially when profiling users of different age ranges. The chosen curse words

were downloaded from the following web pages:

`http://www.hyperhero.com/en/insults.html` and `https://gist.github.com/jamiew/1112488`

Abbreviations

$$AbbreviationUsage = \frac{Abbreviations\_in\_the\_text}{Total\_words} \tag{4.48}$$

This feature is related to the previous one. Abbreviations and Internet slang are commonly used in writings. The tendency of using this type of language can be related to the age of the writer. This resource was created by crawling the list of words provided in `http://onlineslangdictionary.com/thesaurus/words+meaning+acronyms+(list+of).html`.

4.3.5 Morpho-Syntactic Dependency Features

Syntax is the formal description of the principles of sentence structure in a given language. Syntax imposes a certain agreement between words (i.e., subject with verb) and determines the basic principles a sentence needs to meet to be considered structurally correct.

Syntactic analysis is a very important and novel part of our profiling methods. The way authors structure their sentences and their stylistic choices (most of them, unconscious) are automatically analyzed to profile the authors. To do so, we analyze both the word categories used in their texts and the syntactic trees of their sentences. The syntactic features comprise the largest and most effective feature group of our feature set. To create the dependency trees and also obtain the part-of-speech categories of each word, a joint part-of-speech tagger and dependency parser is used, namely (Bohnet and Nivre, 2012).

In the next sections, we introduce the part-of-speech concept, the chosen syntactic paradigm, the sentential dependency structures and the analysis and feature extraction process that is performed to use them.

4.3.5.1 Parts of Speech

The specific part-of-speech tagset used in the experiments is the set of the Penn Treebank Project. This choice is motivated by several reasons. Firstly, it is a precise, fine-grained tagset that does not only distinguish between basic part-of-speech tags (such as “noun”, “verb”, etc.), but also gives information about the specific type of category in question (indicating, for instance, that an adverb is comparative, in which tense a verb is, whether a noun is in singular/plural, or

whether it is common or proper). The PennTreebank tagset provides much more information than the basic tagsets frequently used in the literature. Moreover, this tagset is also used in many widely distributed NLP tools such as CoreNLP, openNLP or the Natural Language Toolkit (NLTK).

The tags and their description are displayed in Table 4.3.

For each word in a text, the parser outputs[3] the corresponding part-of-speech tag. Using this information, we measure the frequency of each of these tags (dividing the number of occurrences of a particular part-of-speech tag by the total number of words in the text). To complement these fine-grained tag frequencies, the frequencies of basic part-of-speech categories (verbs, nouns, adverbs, adjectives, pronouns, determiners and conjunctions) are also computed. In addition, the usage ratios of superlative/comparative adjectives/adverbs as well as verbs in past and present tense (with respect to the total number of verbs) are computed.

4.3.5.2 Syntactic Dependency Trees

Syntactic dependency trees are unordered rooted trees that represent the syntactic structure of a sentence according to a specific grammar. Dependency trees are composed of sets of nodes which correspond to the words of the represented sentence and sets of arcs that connect the nodes via binary asymmetrical dependencies. Each word (except the root) can govern or be governed by another word.

Robinson (1970) formulates the four basic axioms that a syntactic dependency structure must meet to be considered well-formed. These axioms are the following:

1. One and only one element (the root) is independent.

2. All others elements depend directly on some element.

3. No element depends directly on more than one other element.

4. If A depends directly on B and some element C intervenes between them (in the linear order of the string), then C depends directly on A or B or some other intervening element.

The dependency relations that connect two nodes of the tree express syntactic characteristics of that part of the sentence.

According to Mel'čuk, the father of the "Meaning-Text Theory" (MTT)[4], "It has been shown that a dependency tree is closer to a semantic representation and

[3]Note that the chosen parser is a joint tagger-dependency parser

[4]For more information about the MTT, refer to, for instance, (Kahane, 2003)

Tag	Description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP$	Possessive wh-pronoun
WRB	Wh-adverb

Table 4.3: Penn Treebank part-of-speech tagset description.

that dependency-based approaches are more adapted to processing of valency constraints and of multi-word expressions and particularly lexical functions" (Mel-

cuk, 1988). In Mel'čuk's theory, seven levels of language representation are considered, with six sets of rules that map the structures of one level onto structures of adjacent levels. Figure 4.4[5], shows the seven levels of representation considered by the MTT.

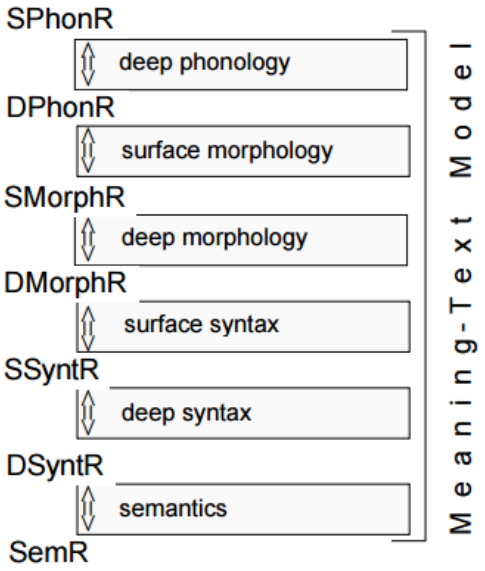


Figure 4.4: MTT representation levels.

The syntactic representations in the MTT are dependency trees at both SSyntR and DSyntR levels. We choose the SSyntR level for several reasons. Thus we believe that dependency trees are much more informative than constituency-based trees. Each dependency has a specific role and the usage of each one of these dependencies can grant our models much more information than constituency-based trees can. Furthermore, the mappings to higher or lower levels of representation of the MTT model can also be used to profile the authors. Although for now only surface-syntactic structures have been used, deep-syntactic structures have been considered and will be used in the future.

Dependency relations provide useful information about the inner structure of the sentences: we can easily measure the tendency of an author to use coordinate or subordinate clauses, the usage of appositions, logical subjects in the passive voice, or the usage of verb chains. Each dependency reveals valuable information about the stylistic choices of the authors. The particular set of dependencies that is used throughout this thesis corresponds to the Penn Treebank Project's dependency relations (described in (Surdeanu et al., 2008)). These dependencies can be

[5]Source: `http://www.neuvel.net/meaningtext.htm`

divided into atomic (single label) and non-atomic (multiple labels) dependencies. The atomic dependencies are presented in Table 4.4

Apart from the atomic dependencies, which are the most common, non-atomic dependencies are also derived by the parser. This type of dependencies is a combination of more than one dependency; cf. Table 4.5.

Each occurrence of "GAP" indicates that the linguistic phenomenon of "gapping" is present. Consider a simple example to understand gapping. In sentences such as "I will buy a car and you ∅ a motorbike", where there is clearly an element missing (in this case, ∅ = *will buy*), a dependency that starts with "GAP" indicates that apart from the relation expressed by the second part of the dependency name, there is a gap: an element that does not appear explicitly in the tree but that has been detected by the parser. Other non-atomic dependencies indicate that there exists a combined relation (the two words are united by both dependencies at the same time), e.g., temporal and manner relation, predicative complement and temporal relation, etc.

For each dependency, a feature is computed. To do so, we divide the number of times that the dependency is used in a text with the total number of sentences. These frequencies correspond to the mean number of the occurrences of each dependency relation per sentence. For example, a high number of "SUB" and "COORD" relations per sentence could indicate that the author tends to use coordinate and subordinate clauses frequently in his/her writings.

4.3.5.3 Tree Complexity measures

To further characterize the writings of the authors, shape-based tree features are computed. Three different metrics are extracted from the trees: depth, width and ramification factor. Depth is defined as the maximum distance (in terms of nodes) between the root and a leaf node. Width is the maximum number of siblings in a level of a tree. Ramification factor is the mean number of children nodes per level.

In the tree shown in Figure 4.5, the maximum width is 4, the maximum depth is 5 and the ramification factor is 3 (4+4+3+1 children per level/ 4 levels).

Each feature is calculated by dividing width, depth and the ramification factor respectively by the number of sentences in a text (which corresponds to the number of dependency trees). This is a way to see how complex the sentences of an author are in a given text. These three features have been very effective in many experiments, which are presented in Chapter 5.

In addition to computing these metrics for the general tree structure, we compute them also for the subordinate and coordinate clauses. Merely the presence of these clauses is already valuable information, which is complemented by the complexity measures.

Tag	Description
ADV	General adverbial
AMOD	Modifier of adjective or adverbial
APPO	Apposition
BNF	Benefactor complement in dative shift
CONJ	Second conjunct (dependent on conjunction)
COORD	Coordination
DEP	Unclassified
DIR	Adverbial of direction
DTV	Dative complement in dative shift
EXT	Adverbial of extent
EXTR	Extraposed element in cleft
HMOD	Token inside a hyphenated word (dependent on the head of the word)
HYPH	Token part of a hyphenated word (dependent on a preceding part)
IM	Infinitive verb (dependent on infinitive marker to)
LGS	Logical subject of a passive verb
LOC	Locative adverbial or nominal modifier
MNR	Adverbial of manner
NAME	Name-internal link
NMOD	Modifier of nominal
OBJ	Object
OPRD	Predicative complement of raising/control verb
P	Punctuation
PMOD	Modifier of preposition
POSTHON	Posthonorific modifier of nominal
PRD	Predicative complement
PRN	Parenthetical
PRP	Adverbial of purpose or reason
PRT	Particle (dependent on verb)
PUT	Complement of the verb put
ROOT	Root
SBJ	Subject
SUB	Subordinated clause
SUFFIX	Possessive sufix (dependent on possessor)
TITLE	Title (dependent on name)
TMP	Temporal adverbial or nominal modifier
VC	Verb chain
VOC	Vocative

Table 4.4: Atomic dependency relations.

Relation
ADV-GAP
AMOD-GAP
DEP-GAP
DIR-GAP
DIR-OPRD
DIR-PRD
DTV-GAP
EXT-GAP
EXTR-GAP
GAP-LGS
GAP-LOC
GAP-LOC-PRD
GAP-MNR
GAP-NMOD
GAP-OBJ
GAP-OPRD
GAP-PMOD
GAP-PRD
GAP-PRP
GAP-PUT
GAP-SBJ
GAP-SUB
GAP-TMP
GAP-VC
LOC-MNR
LOC-OPRD
LOC-PRD
LOC-TMP
MNR-PRD
MNR-TMP
PRD-PRP
PRD-TMP

Table 4.5: Non-atomic dependency relations.

4.3.6 Discourse features

Rhetorical Structure Theory (RST) is a linguistic theory formulated by William Mann and Sandra Thompson in 1988; cf. (Mann and Thompson, 1988). It is a descriptive linguistic approach to the organization of discourse.

RST addresses text organization by connecting parts of a text via rhetorical re-

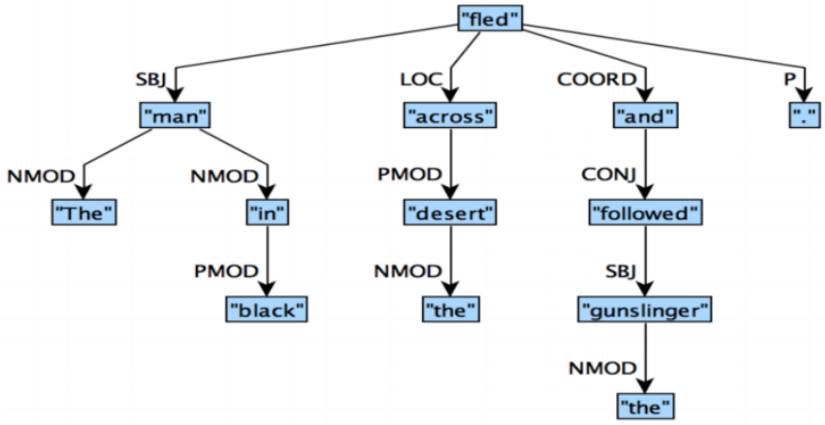


Figure 4.5: Dependency tree example.

lations. Rhetorical relations can also be called "coherence relations" or "discourse relations".

The RST represents whole texts as discourse trees. The discourse trees are hierarchical connected structures where every part has a role with respect to other parts of the text. These parts are also called "spans" or "elementary discourse units". A span, can be composed of one or more text fragments.

RST relations are applied recursively to the input text until all spans in the text are parts of RST relations and, as a result, parts of a discourse tree.

Discourse trees have the following characteristics:

- **Completeness:** One relation dominates the whole tree.
- **Connectedness:** Each token in a text is in a span, and each span is connected with one or more spans.
- **Uniqueness:** Each relation connects different sets of spans. Each span can only be a part of one relation.
- **Adjacency:** The spans that are connected in a relation constitute a contiguous text span.

An example of an RST tree is shown in Figure 4.6. Each arc is labeled by the name of a relation, which connects spans.

In a relation, the connected spans can have two different roles: nucleus or satellite[6]. A nucleus of a relation is the text span that is considered the crucial

[6]Disclaimer: even though this characteristics of the discourse relations are not used in the development of the presented discourse features, we plan on expanding them to consider satellite-nucleus configurations. For more information about our future work, see Chapter 6.

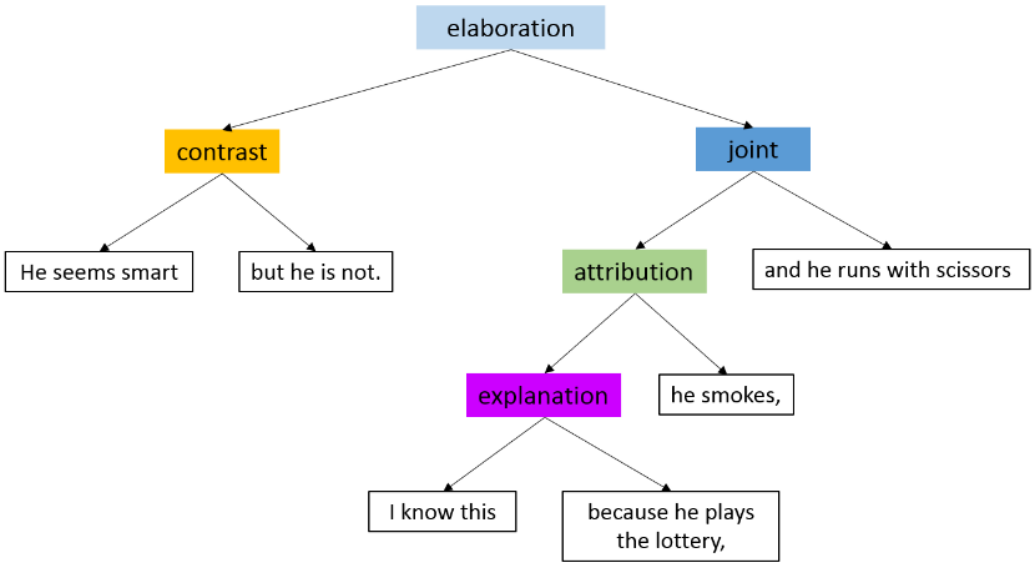


Figure 4.6: Discourse tree example.

part of the relation, whereas satellite nodes are complementary: they contribute to the nuclei, but they are not the key part of the relation. Two types of relations are observed: nucleus-satellite and nucleus-nucleus (multinuclear). The first case was just described; in the latter, all connected text spans are equally important.

To illustrate both types of relations, some relation examples for both configurations are provided in Table 4.6[7].

Nucleus-Satellite Relations		
Relation Name	**Nucleus**	**Satellite**
Background	Text whose understanding is being facilitated	Text to facilitate understanding
Elaboration	Basic information	Additional Information
Evaluation	A situation	Evaluative comment on the situation
Evidence	A claim	Information to increase reader's belief in claim
Summary	Text	Short Summary of that text
Multinuclear Relations		
Relation Name	**Nucleus**	**Nucleus**
Contrast	One alternative	The other alternative
List	An item	Another Item

Table 4.6: RST relation examples.

As we can see, RST relations are used to characterize the text structure at a high level. The analysis of the usage of these relations can help characterize the

[7]Source: `http://www.sfu.ca/rst/01intro/definitions.html`

writings of an author.

Using discourse features proves to be very useful to extract stylistic patterns and profile the authors.

RST has been used in several fields such as text generation and summarization.

The application of the Rhetorical Structure Theory to author profiling and author identification is a novel and effective approach to the characterization of the writing style of authors. To apply the theory, a discourse parser is used, namely (Surdeanu et al., 2015), which divides the input text into elementary discourse units and links them via discourse relations. The full set of discourse relations that is used by the parser is the following: Joint, Background, Condition, Evaluation, Summary, Cause, Contrast, Topic-comment, Elaboration, Comparison, Topic-change, Textual-organization, Enablement, Attribution, Explanation, Same-unit and Manner-means.

The features that are extracted from the discourse trees are the frequencies of each discourse relation per elementary discourse unit and the shape metrics of the trees used before: depth, width and ramification factor. A high amount of topic changes per discourse unit can be seen as a chaotic structure of the text. On the other hand, a text that is mainly a set of elementary discourse units linked by Elaboration relations is a fairly straightforward text. Each discourse relation provides thus interesting characteristics of the textual organization and coherence of the authors' writing. The shape metrics analyze the complexity of the discourse structure, which are useful complementary features to further characterize the authors writing.

4.4 Feature Vector Construction

After introducing the developed feature set, the transition from feature extraction to feature vectors needs to be discussed.

Each feature value is computed for each instance of the dataset. Each text in the dataset is an instance and is represented by a corresponding feature vector. Each one of the features of an instance is assigned to a specific dimension of the corresponding feature vector. It is very important for every vector to have the same length and for each feature to occupy the exact same dimension among all vectors. In cases where a specific feature is not available for an instance (e.g., in the syntactic group, there might be dependency relations that do not appear in a specific instance), 0.0 is introduced.

These vectors and their labels are the input for the chosen classifier. To maximize the performance of the classifier, normalization is sometimes required. The main transformation that is applied to our feature vectors is "Standarization" (especially when using SVM algorithms for classification). This transformation

rescales the data to have mean of 0.0 and standard deviation of 1.0. The basic formula applied in these cases is the following:

$$X_{st} = \frac{X - \mu}{\sigma} \tag{4.49}$$

In 4.49, X_{st} is the standardized value of a feature value of a specific vector, X is its original value, μ is the mean value of this feature among all vectors, and σ is the standard deviation of this feature among all vectors.

After this process, the classifier can start searching for patterns and learn from the provided labeled data to classify unseen instances.

4.5 Feature Set Evaluation

In this section, we compare our feature set with commonly used features in the state of the art in order to prove the relevance of our approach. As stated before, many approaches rely mainly on lexis instead of structure. We argue that syntactic and discourse structures are very powerful profiling means that have been underestimated in the related work. To prove that our hypothesis is correct, we compare the performance our feature set with the performance of the following sets of features: function/stop words, part-of-speech usage, token n-grams and bag-of-words feature sets. In every case, the same classifiers are used with our approach and the alternative feature set.

4.5.1 Function/Stop words and Part-of-speech

Function/stop words and part-of-speech frequencies are features that have been used in author profiling and identification for a long time (see Chapter 3).

To compute the stop/function word features, lists of stop/function words are required. The feature vectors that are usually created are composed of the usage ratio of each considered function/stop word in a text (the dimensionality of these vectors depends on the number of function/stop words considered). The PoS feature set creates feature vectors in which each dimension indicates the percentage of words of a text that correspond to a specific morpho-syntactic category.

To compare our set of features with function/stop words and part-of-speech features, we use two of the resources that have been created during the development of this thesis, namely the "AuthorshipDat" and the "LiteraryAmerican" corpora (more information about the characteristics of each resource can be found in Section 4.2). Recall that the first corpus is composed of journalistic blog posts from British newspapers, written by 23 authors. The second dataset contains the

writings of 16 different American novelists. Both datasets are labeled by the gender and author of each text. Both problems are multi-class classification problems (23-class and 16-class respectively). For classification, SVMs with a linear kernel and 10-fold cross-validation are used.

Tables 4.7 and 4.8 show the performance of our feature set (displaying the performance of the full set and each feature group separately) compared to the mentioned baseline feature sets (function/stop word usage, PoS frequency and combinations of them) in the case of gender and author identification on both datasets.

Features Used	**Accuracy AuthorshipDat**	**Accuracy LiteraryAmerican**
Full Set	**78,16%**	**91,08%**
Character-based	62.18%	76.05%
Word-based	51.72%	65.79%
Sentence-based	19.84%	29.36%
Dictionary-based	27.84%	68.53%
Syntactic	68.44%	91.01%
Discourse	40.11%	64.39%
Function Words (FW)	66.99%	86.36%
Stopwords (SW)	65.81%	85.85%
Parts of Speech (PoS)	63.30%	81.01%
FW + PoS	72.45%	86.96%
SW + PoS	72.78%	87.29%

Table 4.7: Results of the author identification experiments on both datasets.

Features Used	**Accuracy AuthorshipDat**	**Accuracy LiteraryAmerican**
Full Set	**89,97%**	**90,71%**
Character-based	87.91%	81.02%
Word-based	81.18%	78.79%
Sentence-based	65.01%	73.88%
Dictionary-based	71.45%	84.39%
Syntactic	85.17%	90.76%
Discourse	75.34%	75.22%
Function Words (FW)	81.72%	52.73%
Stopwords (SW)	81.46%	52.73%
Parts of Speech (PoS)	81.53%	74.84%
FW + PoS	82.67%	76.81%
SW + PoS	82.88%	76.87%

Table 4.8: Results of the gender identification experiments on both datasets.

Several conclusions can be drawn from the tables. First, our full set of features outperforms the alternative feature sets and their combination. In some cases, individual feature groups are able to outperform the alternative sets of features, more specifically, the syntactic group of features seems to be the best performing individual group in the majority of the cases, proving the relevance of syntax in these experiments. Furthermore, it is interesting to note how the function word approach achieves very competitive accuracy when identifying the authors of literary texts, but fails to do so in the gender identification experiment on the same dataset. This is probably due to the highly stylistic content of the literary texts, where clear patterns in function word usage can be found with respect to their author, but not with respect to the gender of their author.

The PoS feature set achieves competitive performance in all cases, which is one of the reasons why this particular set of features forms part of our feature set.

4.5.2 Token n-grams

We compare our feature set with the token n-gram feature set using the LiteraryBritish corpus. The 100, 300, 500, 700 and 900 most frequent token 2-5 grams are considered for gender and author identification.

The results of our feature set (including the performance of the full set and every individual feature group) are shown in Table 4.9. The results of the token n-gram feature set are shown in Table 4.10.

Used Features	**Accuracy Gender**	**Accuracy Author**
Complete Set	**90.18%**	88.34%
Character-based	67.65%	37.76%
Word-based	61.79%	38.54%
Sentence-based	60.35%	17.12%
Dictionary-based	60.62%	17.90%
Syntactic	88.94%	82.82%
Discourse	69.99%	42.61%

Table 4.9: Results of our features in the gender and author identification experiment.

The tables show that the full set of features outperforms every n-gram combination. However, the accuracy achieved by the n-gram feature set in the gender identification case is very competitive and outperforms all the individual feature groups, except the syntactic features. This also happens in the author identification experiment. We can also observe that the token n-gram feature set is much more effective when applied to a relatively simple problem such as gender identification than to an 18-class author identification scenario. It is worth noting that every

N	#ngrams	Accuracy Gender	Accuracy Author
2	100	84.60%	63.97%
2	300	84.32%	64.19%
2	500	84.66%	64.58%
2	700	84.61%	62.46%
2	900	83.32%	63.52%
3	100	76.85%	40.04%
3	300	78.58%	42.61%
3	500	77.41%	44.11%
3	700	79.64%	46.23%
3	900	79.86%	45.11%
4	100	68.20%	30.11%
4	300	72.61%	34.96%
4	500	75.68%	39.15%
4	700	75.90%	38.59%
4	900	76.35%	39.48%
5	100	61.01%	23.42%
5	300	69.26%	26.77%
5	500	71.38%	27.49%
5	700	71.16%	27.99%
5	900	71.55%	28.16%

Table 4.10: Results of the n-gram feature set.

combination, except the 100 n-gram combination, uses a much higher amount of features than our feature set.

4.5.3 Bag of words

To compare our approach to a bag-of-words approach, first of all, the most frequent N words are extracted and the frequency of these words in each text is used as features. As in Section 4.5.1, two datasets are used: the LiteraryBritish dataset and the AuthorshipDat corpus, such that we can see the performances in two different genres. Gender and author identification are performed on both datasets. Table 4.11 shows the results of our feature set on both datasets (showing the performance of the full set as well as of each individual feature group separately).

Table 4.12 shows the performance of the bag-of-words approach using the most frequent 100, 300, 500, 700 and 900 words on both datasets and both classification problems.

Two conclusions can be drawn: the bag-of-words approach outperforms our

Features Used	Accuracy Gender LiteraryBritish	Accuracy Author LiteraryBritish	Accuracy Gender AuthorshipDat	Accuracy Author AuthorshipDat
Full Set	**90.18%**	**88.34%**	**89.97%**	**78.16%**
Character-based	67.65%	37.76%	87.91%	62.18%
Word-based	61.79%	38.54%	81.18%	51.72%
Sentence-based	60.35%	17.12%	65.01%	19.84%
Dictionary-based	60.62%	17.90%	71.45%	27.84%
Syntactic	88.94%	82.82%	85.17%	68.44%
Discourse	69.99%	42.61%	75.34%	40.11%

Table 4.11: Results of our features in the LiteraryBritish and AuthorshipDat corpus.

#words	Accuracy Gender LiteraryBritish	Accuracy Author LiteraryBritish	Accuracy Gender AuthorshipDat	Accuracy Author AuthorshipDat
100	91.74%	86.17%	83.75%	72.40%
300	91.91%	87.23%	87.51%	77.68%
500	90.68%	87.39%	87.81%	78.15%
700	91.13%	86.33%	87.44%	77.40%
900	90.18%	84.82%	88.00%	77.45%

Table 4.12: Performance of the bag-of-words approach.

features in the gender identification case, but it is outperformed (by a small margin) in the author identification case. As we can see, the performance of this approach is very competitive and outperforms each individual feature group. Given the competitive performance of this kind of features, an interesting test is to combine these lexical features with our feature set, in order to see whether the combination of structural and lexical features results in a performance improvement.

Table 4.13 shows the performance of our features (NoLex) compared with the combination of our features and the frequency of the most common 100, 300, 500, 700 and 900 words (Full100, Full300, Full500, Full700 and Full900 respectively), on both datasets and problems.

Features Used	Accuracy Gender LiteraryBritish	Accuracy Author LiteraryBritish	Accuracy Gender AuthorshipDat	Accuracy Author AuthorshipDat
NoLex	90.18%	88.34%	89.97%	78.16%
Full100	94.92%	93.31%	91.03%	87.37%
Full300	94.59%	92.19%	91.24%	88.28%
Full500	94.75%	92.97%	91.10%	87.51%
Full700	93.25%	92.74%	90.59%	87.44%
Full900	93.08%	91.57%	90.66%	87.09%

Table 4.13: Performance of our system combined with lexical features.

We can observe that the combination of lexical features and our feature set

always results in a better performance: in each case, the combination of features outperforms both our features and the bag-of-words feature sets. So, even if the classification with our features is sometimes outperformed by bag-of-words features, combining both sets of features results in better accuracy.

All comparisons discussed above, tell us that our feature set is very effective. It outperforms most of the state-of-the-art classic feature sets and also complements them.

4.6 Conclusions

In this chapter, the full set of features proposed in the thesis has been introduced. This set of features has also been compared with alternative feature sets that have been commonly used in the related work. All experiments that will be presented in Chapter 5 use these features.

To sum up and to have an overview of the complexity of the feature set, consider Table 4.14.

Feature Group Name	**Identifier**	**Number of features**
Character-Based	CB	13
Word-Based	WB	11
Sentence-Based	SB	3
Dictionary-Based	DBDisc	1
	DBInt	1
	DBPol	2
	DBMood	17
	DBCurse	1
	DBAbbrev	1
Syntactic	SynF (dependency frequencies)	69
	SynS (shape metrics)	9
	SynP (Part of Speech)	49
Discourse	DiscF (relation frequencies)	17
	DiscS (shape metrics)	3
Total Number of features		197

Table 4.14: Feature summary.

Table 4.14 shows the feature groups that were introduced and the number of features per group. Each group (and in some cases, relevant subsets of the group) is assigned an identifier. We will use these identifiers when introducing the experiments that were performed to refer to the particular features used in those

experiments.

For more information about the limitations and future improvements of the feature set, refer to Chapter 6.

Chapter 5

APPLICATIONS

In this chapter, all experiments and applications of our approach that have been implemented in the context of the thesis are outlined. The applications are categorized into supervised and unsupervised/semi-supervised learning approaches.

The main goal behind each experiment is to prove the relevance, versatility and profiling potential of our deep-linguistic feature set. Choosing a small set of deep-linguistic features proves to be a very effective strategy, and it is shown that less than 200 features can perform at state-of-the-art level in many different scenarios.

The applications use different text types, ranging from opinion columns and blog posts to literary texts. The majority of works use texts written in English, but French, Italian, Spanish, Catalan and German are also considered. The tackled problems include gender, age, language, geographic origin and author identification.

5.1 Supervised Approaches

The majority of the experiments that have been implemented in the context of this thesis involve supervised learning approaches. These approaches usually apply 10-fold cross validation and use Python and its Natural Language Toolkit to extract the majority of the features. As already mentioned, dependency and discourse trees are built using (Bohnet and Nivre, 2012) as dependency parser and (Surdeanu et al., 2015) as discourse parser, respectively. SVM and Random Forests are the most often used classifiers, more specifically Weka's implementations of both algorithms.

5.1.1 Overview

Several supervised learning experiments have been implemented. We initially focused on the identification of the gender of the authors. To do so, the first version of the feature set presented in Chapter 4 was developed. These features were derived from an empirical study of a development corpus sample following the assumption that they are the most distinctive for the writing styles of women and men. Age identification was also considered in the first set of experiments. To perform age identification, a set of features especially selected for the task was considered.

After successfully proving that the first version of the feature set was effective in gender identification using texts written in English, we wanted to prove that the approach was competitive in several different languages. Language identification was also performed using structural features.

Discourse and further syntactic features (PoS frequencies and shape measures to coordinate/subordinate clauses) were added to the feature set and used to perform gender identification on the same corpus as in the previous experiments, outperforming them and showing that the full set of features is more effective.

Author identification experiments are then introduced. The first experiment uses opinion pieces and the first version of our feature set to distinguish between the writings of 26 authors. Then, we combine both types of experiments and show that our features are effective in gender and author identification with literary texts and blog posts. Finally, we expand these experiments to identify gender, geographic origin, the book to which a selected piece of writing belongs and author.

In each of the presented experiments, the system is able to achieve very competitive performance.

Let us now introduce the applications of our approach that use supervised learning in detail.

5.1.2 Gender and Age Identification using Blog Posts

Gender and age identification on a corpus of blog posts has been the first set of experiments that we implemented. For these experiments, the NYTimes dataset has been compiled (see Chapter 4). A set of 82 features (syntactic dependency features constitute the biggest share (67) of them), and Weka's Random Forests implementation for classification are used.

5.1.2.1 Features

Five different types of features are used. The combination of features we use aims to capture the writing of men and women from the most basic level (usage of characters) to a more global level (sentence structure).

The selected features are the following: Character-based (CB), Word-based (WB), Sentence-based (SB), Dictionary-based (in this case, only the DBpol features are used) and Syntactic (SynF and SynS).

Table 5.1 displays the number of features of each type we used in these experiments.

Feature Category	**#Features**
Character-based	6
Word-based	5
Sentence-based	2
Dictionary-based	2
Syntactic	67

Table 5.1: Distribution of features across categories.

As pointed out in Chapter 3, many works in author profiling and author gender identification use dictionaries to analyze the content of the texts. The novelty here is to use polarity dictionaries to measure the expressiveness of the authors and use this information to distinguish between genders. This is a technique that is used more often in Sentiment Analysis.

Dictionary-based and syntactic features capture the expressiveness of a text and the syntactic stylistic idiosyncrasies. These groups of features prove to be very effective for gender identification.

5.1.2.2 Experimental Setup

As already mentioned above, we use Weka's Random Forests implementation. The output of the feature extraction is represented as an Arff file, in which all the texts are represented in terms of multi-dimensional vectors, with each feature as a separate dimension and one of the values of a feature as instantiation of its dimension.

The Arff file is fed to Weka for classification. To obtain reliable performance figures, we use a 10-fold cross validation process, such that the outcome of the classification does not depend on which specific part of the dataset was used for training and which part for testing.

To explore the relevance of the different feature types both in combination with other features and in isolation, we run a number of experiments, each of them with a specific feature set; see the first column of Table 5.1 for the different feature sets that we use in our experiments.

Table 5.2 lists the accuracy figures obtained on this dataset when using different subsets of our feature set.

Feature combination	**#Features**	**Accuracy (%)**
Sentence-based (S)	2	56.81%
Dictionary-based (D)	2	59.75%
S + D	4	60.59%
Word-based (W)	5	63.63%
Character-based (C)	6	64.53%
C + S	8	64.71%
W + C	11	66.45%
C + S + D	10	66.63%
C + D	8	66.99%
W + D	7	67.46%
W + S + D	9	68.18%
W + C + S + D	15	69.86%
W + C + D	13	70.28%
Syntactic (Y)	67	77.03%
Y + D	69	77.39%
Y + W	72	77.87%
Y + S	69	78.35%
Y + S + W	74	80.32%
Y + C	73	81.16%
Y + C + W	78	82.12%
Y + C + S	75	82.35%
Y + C + S + W + D	82	82.72%
Y + C + S + W	80	82.83%

Table 5.2: Performance of our approach on the NY Times blog dataset when using different feature sets.

In order to assess the relevance of our feature set, we carry out an additional experiment with a totally different approach, using bag-of-word features. The individual posts are thus considered as vectors where each dimension stands for the percentage of the occurrence of a specific frequent word in them. To obtain the set of frequent words, we discard stop words and calculate the tf-idf measure

for all the remaining words. The 1,000, 2,000 and 3,000 words with higher tf-idf values are used for classification. The experiment is run on the same dataset; the classifier is again Weka's implementation of Random Forests. The results of this experiment are summarized in Table 5.3.

#Features	**Accuracy (%)**
1,000	66.09%
2,000	72.49%
3,000	73.80%

Table 5.3: Performance of the bag-of-words approach on the NYTimes dataset.

5.1.2.3 Discussion

Table 5.2 shows that using the whole set of features, we obtain an accuracy of 82.72%. This is an accuracy that is definitely within the range of the accuracies achieved by the state-of-the-art approaches in this area. It is also remarkable that this accuracy has been achieved using a much smaller number of features than in most of the state-of-the-art approaches. It is also important to highlight that using only 13 features, an accuracy of 70.28% is achieved. Table 5.2 furthermore shows that the use of syntactic dependency features pays off. Using only this group of features, we achieve an accuracy of 77.03%. This gives us a hint that there are important differences in how men and women syntactically structure their sentences.

In contrast, when only content features are used in a bag-of-words approach, the performance decreases significantly (see Table 5.3), despite the enormous increase of the number of features. In other words, the use of mere lists of words implies significantly more complex feature management and leads to worse performance than the use of context independent structural sentence features.

5.1.2.4 One Step further to Author Profiling

After proving that our feature set leads to a good performance for gender identification, we explore whether it can equally be used for other demographic traits such as age. For this purpose, we use another collection of blog posts as dataset. This dataset is described in (Schler et al., 2006) and is also used by Argamon et al. (2009). It is composed of informal blog posts extracted from blogger.com. The blogs posts are labeled by the gender and the age of the author and are thus ideal for our experiments.

The ages are grouped into three classes: “teens”, which represents the authors whose age ranges from 13 to 17, “twenties”, which goes from age 23 to 27, and “thirties”, which captures the authors who are older than 30. In contrast to the NYTimes posts, these blogs are not well structured and written. They contain many orthographic errors, slang expressions, abbreviations, emoticons, spam content, texts that are only composed of pasted music lyrics, etc.

We perform two different classification runs: one in which the classifier predicts whether the author is a man or a woman, and the other that determines in which of the three age classes the author is situated. These classification runs are performed using a balanced subset of the dataset that is composed of 5,955 posts. The classifier that is used in these experiments was SMO, a variant of Support Vector Machines with a radial kernel.

Using the same set of features, we obtain the figures outlined in Table 5.4 (as baseline, we use the accuracy of the majority class classifier):

	Gender	**Age**
Accuracy	68.09%	55.96%
Baseline	50%	33%

Table 5.4: Performance of our approach with the same set of features when classifying by gender and age.

Although the performance of gender identification is here considerably lower than the one we achieved on the NYTimes corpus, we see that age classification improves over the baseline by 22,96%. The analysis of the results further reveals that due to the numerous orthographic and syntactic errors encountered in the dataset, the performance of (Bohnet and Nivre, 2012)’s dependency parser, which is used in our experiments, decreases significantly. Since the syntactic features constitute the majority of our feature set, our hypothesis is that the poorer performance of the parser is (at least partially) responsible for the lower performance of our system. To tackle this problem, we use a shallow parser that is a simplification of our original dependency parser and that is expected to be more tolerant to faulty texts. Several other features are added to boost the performance in the case of age classification. These features are:

1. number of orthographic errors per word,
2. percentage of discourse markers,
3. frequency of curse words and abbreviations,
4. usage of passive voice,

5. further dictionary based features that measure the usage of words related to school, college, duties and leisure time.

The total number of features that are used in this extended dataset is 100. After the new features are introduced, the performance of the classification is as shown in Table 5.5:

	Gender	**Age**
Accuracy	66.97%	62.92%
Baseline	50%	33%

Table 5.5: Performance of our approach when classifying by Gender and Age with an extended feature set.

As we can observe, the introduction of these new features leads to a slight decrease of accuracy in gender identification, but, at the same time, to a considerable increase in age classification. It is obvious that the quality of texts influences the performance of author profiling and that in order to capture idiosyncratic features of a specific genre (such as recurrent orthographic and syntactic mistakes), specific features must be introduced.

5.1.2.5 Conclusions

The most obvious conclusion from these experiments is that a collection of distinctive features that characterize blog postings – including traditional word-oriented features, but also sentential features, marked "positive" and "negative" words (as used in sentiment analysis), etc. – help distinguish between writings of men and women. The differences between these writings can be observed at many different levels. Thus, we can see differences in how specific characters, words and sentences are used, as well as in how expressive the texts are.

We also demonstrated that there are important differences at the sentence structure level. We compared our system to a bag-of-words approach in which only content-features were used to classify. We saw that using thousands of features that relied only on the content of the texts did not outperform our system, so the conclusion is that there is no need to use thousands of features to obtain a good accuracy for gender identification, if the features are chosen carefully.

We saw that the usage of the same feature set for age identification improved the baseline by more than 20%. With some extra features that were more age-oriented, the accuracy improved the baseline by over 29%. That is, even this small experiment has shown that author profiling using a small set of features is a feasible goal, but that these features must also capture the idiosyncrasies of the authors.

5.1.3 Multiple Language Gender Identification for Blog Posts

After implementing the first experiments on gender identification using texts written in English, we wondered: Does the language background also influence the difference of how men and women write? It is known that an average English sentence has a less complex syntactic structure than a German sentence. Does the assumed difference in the complexity of the syntactic structures in English and German lead to idiosyncrasies in gender identification in English and German?

The vast majority of approaches to data-driven gender identification has been on English material; rather few are on other languages; see, e.g., (Estival et al., 2007) on Arabic, (Rangel and Rosso, 2013b) on Spanish, (Kucukyilmaz et al., 2006) on Turkish and (Pham et al., 2009) on Vietnamese, and there are practically no systematic language-contrastive experiments.

In order to shed some light on the above questions, we carried out three experiments on blog post corpora in Catalan, English, French, German, Italian and Spanish, interpreting the problem of gender and language identification as a supervised classification problem: (i) classification of blog posts in each of these languages with respect to the gender of their authors (man vs. woman); (ii) classification of all posts joined into one multilingual dataset with respect to the gender of the writers; and (iii) classification of all posts with respect to gender and language of the author at the same time (as, e.g., 'male English', 'female Spanish', etc.).

For the first experiment, we use sets of features that are mainly of syntactic nature. For experiments (ii) and (iii), we use strictly language-independent, universal features, such that the classification procedure does not have any explicit language clues. In none of the experiments, content-oriented features (as, e.g., the most common words or n-grams) are used, since content-oriented features let gender identification heavily depend on the training dataset and make it hardly comparable across languages. This makes our proposal different from the vast majority of the state-of-the-art approaches to gender identification, which all heavily draw on content-oriented features (see Chapter 3).

In the next section, the features that are used in the experiments are presented. Then, we describe the experiments and discuss their outcome. Finally, some conclusions from the presented work are drawn.

5.1.3.1 Feature set

The features that are required for cross-language language background studies, as in our case, are features that are entirely or at least to a certain extent (as, e.g., grammatical functions) language- and content-independent. These are structural features. For our experiments, we used four different types of mostly content-

independent features: (i) character-based features (CB), (ii) word-based features (WB), (iii) sentence-based features (SB), and (iv) syntactic features (SynS and SynF. See Chapter 4).

Table 5.6 shows a summary of the number of features of each type that were used.

Type	**# Features**
Character-based Features	15
Word-based Features	14
Sentence-based Features	2
Syntactic Features	22–65

Table 5.6: Feature number overview.

The dependency tag sets differ from language to language and are also of different granularity (from 22 for French to 65 for English). As a result, the number of syntactic features differs from language to language. All the dependency trees were derived using the dependency parser described in (Bohnet and Nivre, 2012), trained on each language.

5.1.3.2 Experimental Setup

For the supervised classification experiments, we used Weka's implementation of a Random Forests classifier.

The features were captured in a file in which all blog posts are represented in terms of multi-dimensional vectors, with each feature as a separate dimension and one of the values of a feature as instantiation of its dimension. To obtain more reliable performance figures, we used 10-fold cross validation, such that the outcome of the classification does not depend on which part of the dataset has been used for training and which part for testing.

5.1.3.3 Datasets

For the compilation of the datasets, the same methodology was used for all six languages (Catalan, English, French, German, Italian and Spanish). We searched for blogs in which the authors were known, such that their gender could be deduced for validation of the performance of our algorithm. For this purpose, we looked for blog sections of online newspapers and magazines. The chosen datasets have been: EnglishDat, FrenchDat, CatalanDat, SpanishDat, ItalianDat and GermanDat (see Chapter 4 for more details).

5.1.3.4 Experiments and their results

We carried out three different experiments, taking as baseline in all three the majority class classifier.

In the first experiment, we carried out gender identification for each language dataset separately. Table 5.7 displays the performance of our classifier in this experiment.

	English	**Spanish**	**German**	**French**	**Catalan**	**Italian**
Accuracy	80.24%	88.02%	77.87%	83.98%	88.11%	86.54%
MajClassBaseline	50%	50%	50%	50%	50%	50%
Number of Features	96	83	73	52	79	52

Table 5.7: Performance of the monolingual gender identification classifier.

For the second and third experiments, the six datasets were merged, such that the resulting dataset is composed of 29,117 texts by male and female authors in Catalan, English, French, German, Italian and Spanish. Furthermore, the set of features was reduced to 27 language-independent features: all punctuation features, the frequency of the usage of acronyms, the frequency of the usage of first person singular/plural pronouns, the frequency of the usage of stop words, the mean number of words per sentence, characters per word, the percentage of words that are more (and less) than 5 characters and the percentage of words that start/end with vowel/consonant.[1]. They are language-independent in the sense that they appear in all of the languages we consider—although they are, obviously, instantiated differently. But since we count only their appearance, not their concrete instantiations, they can indeed be considered universal.

In order to avoid the influence of idiosyncratic characteristics of a language[2] on these features, the feature values are normalized: each value is divided by the value of the corresponding reference feature obtained from a reference corpus of the language in question. As a consequence, we obtain for each text a feature profile that reflects the author's personal writing style rather than a language-inherent bias. Table 5.8 lists the used reference corpora.

In order to be able to normalize features during the experiments, i.e., when we classify a test dataset (and thus do not know the language of a text), we implemented a language prediction procedure. The procedure is based on the similarity

[1] Syntactic features cannot be used here because the dependency relation tag sets are language-specific.

[2] For instance, in German punctuation is much more grammaticalized than in English, where it is highly style-driven. This leads to a higher relative frequency of, e.g., commas and semicolons in German. The same occurs with capitalization: in German, nouns are capitalized.

Language	Corpus
Catalan	Cess_cat
English	Brown
French	Baf
German	Tiger
Italian	Turin university treebank
Spanish	Cess_esp

Table 5.8: Reference corpora.

of the feature values to each of the corresponding reference feature values: the more similar the values, the more likely the language of the reference features is to be used for normalization.

In the second experiment, the texts in the merged dataset were classified with respect to the gender of the authors of the texts. The difference between this experiment and the first one is that in this case the classification is carried out with language-independent features only on a multilingual dataset using feature normalization as described above. The results of this experiment can be seen in Table 5.9.

	Merged Dataset
Accuracy	77.01%
Baseline	50.19%

Table 5.9: Results of multilingual gender identification.

In the third experiment, the texts in the merged dataset were classified with respect to twelve different classes: 'catalan_male', 'catalan_female', 'english_male', 'english_female', etc. The purpose of this experiment has been to assess to what extent we can identify the gender and language of an author in one single dataset analyzing only the writing style of the authors. If this is feasible (again, without any dictionaries or language-dependent features), it can be feasible to identify the native language of an author not only in language learner texts (as usually done in the state of the art; cf. Chapter 3), but also in well-written texts. The results of this experiment are displayed in Table 5.10. The baseline is low because the number classes that are used in this classification process is rather large (recall that we use random classification as baseline).

	Merged Dataset
Accuracy	74.67%
Baseline	12.26%

Table 5.10: Performance of the joint gender and language identification experiment.

5.1.3.5 Discussion

The results of the first experiment show that a set of features that captures mainly the syntactic structure and writing style of an author (rather than the vocabulary and thus content, as does the majority of the state-of-the-art proposals) achieves state-of-the-art accuracy not only, e.g., for English, where such features are more freely used, but also for French, German, etc. where punctuation is much more regularized (such that gender identification is a priori more difficult). The fact that the same features worked very well for all languages can be seen as clear evidence that there are common patterns that distinguish the writing style of both genders for all six languages considered.

The performance figures of the second and third experiments show that a small number of structural features can be used for gender identification with a competitive outcome, and that the writings of the authors of different genders show idiosyncratic patterns of language-independent features that allow for the identification of the language in which they are written. Due to the fact that the use of these patterns by an author is, as a rule, subconscious, it can be hypothesized that it is realistic to assume that it is feasible to identify the gender and native tongue of the author when he or she writes in a foreign language. The hypothesis would be that the writers carry their writing style from their native language to their writings in a foreign tongue.

Figures 5.1 and 5.2 show the contribution of the individual features to the writing style of both genders in our six languages. Each axis represents the normalized mean value of a feature for men and women. Figure 5.1 shows the contribution of the punctuation features, while Figure 5.2 captures the word-oriented features. Remember that the normalized features are calculated as the division between actual feature values and the reference ones. Both graphs have the mean values of the features represented in a logarithmic scale.

Both figures reveal there are several differences between languages at a punctuation and word level, and these differences are what makes both gender and gender and language identification possible. In Figure 5.1, the main differences are in the use of quotation marks of German writers relatively to the other languages. There are also some deviations in the writings of Italian men and women

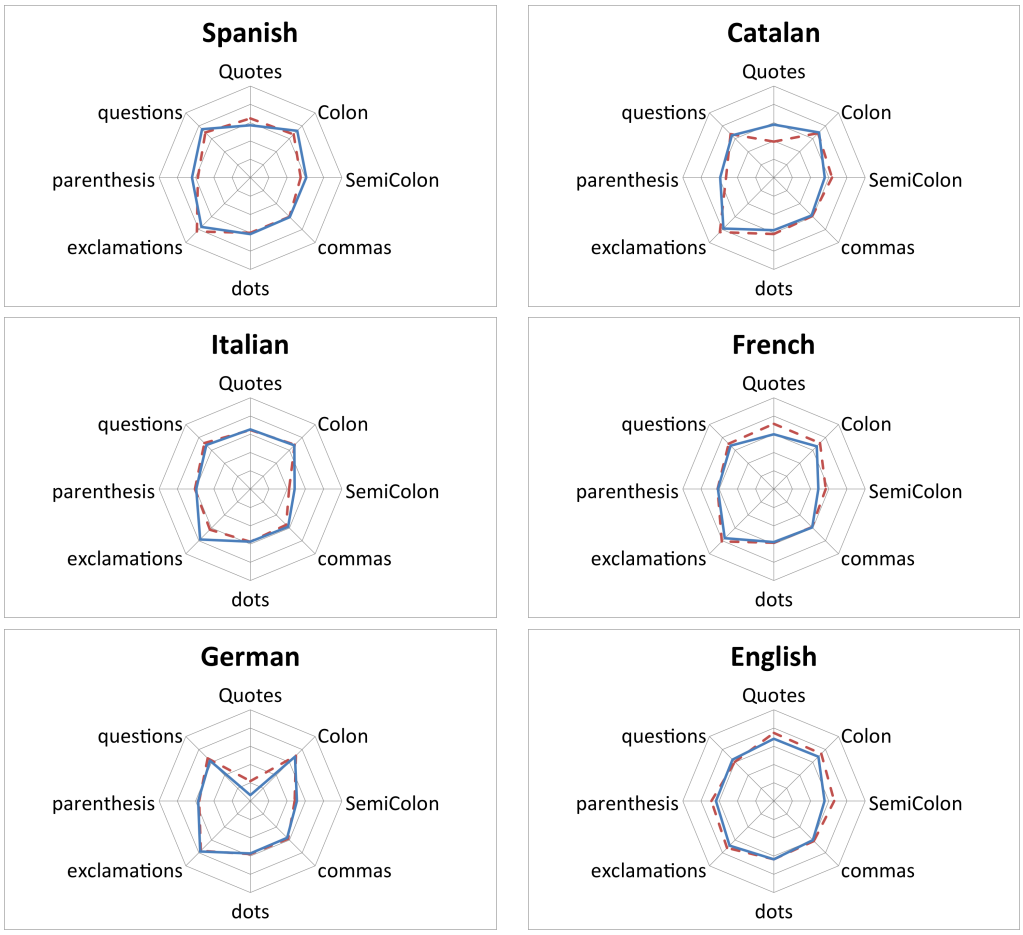


Figure 5.1: Distribution of punctuation features in the posts of men and women across languages; solid line (male), dotted line (female).

with respect to the use of exclamation marks.

In Figure 5.1 it is revealing to compare Spanish and Catalan. Even though both languages are quite similar, we see that the way men and women deviate from the reference features in both languages is different. The deviation in the usage of quotation marks, semicolons, question marks and periods is quite different if we compare the writings of the opposed genders. We can also see that French women deviate more than men in all punctuation features.

German authors are the most different: the values of the features of German authors are smaller than in the other languages in both cases. This means that the deviation from the reference features in German authors is smaller than in the other languages. We can hypothesize that this could be due to the cultural influences.

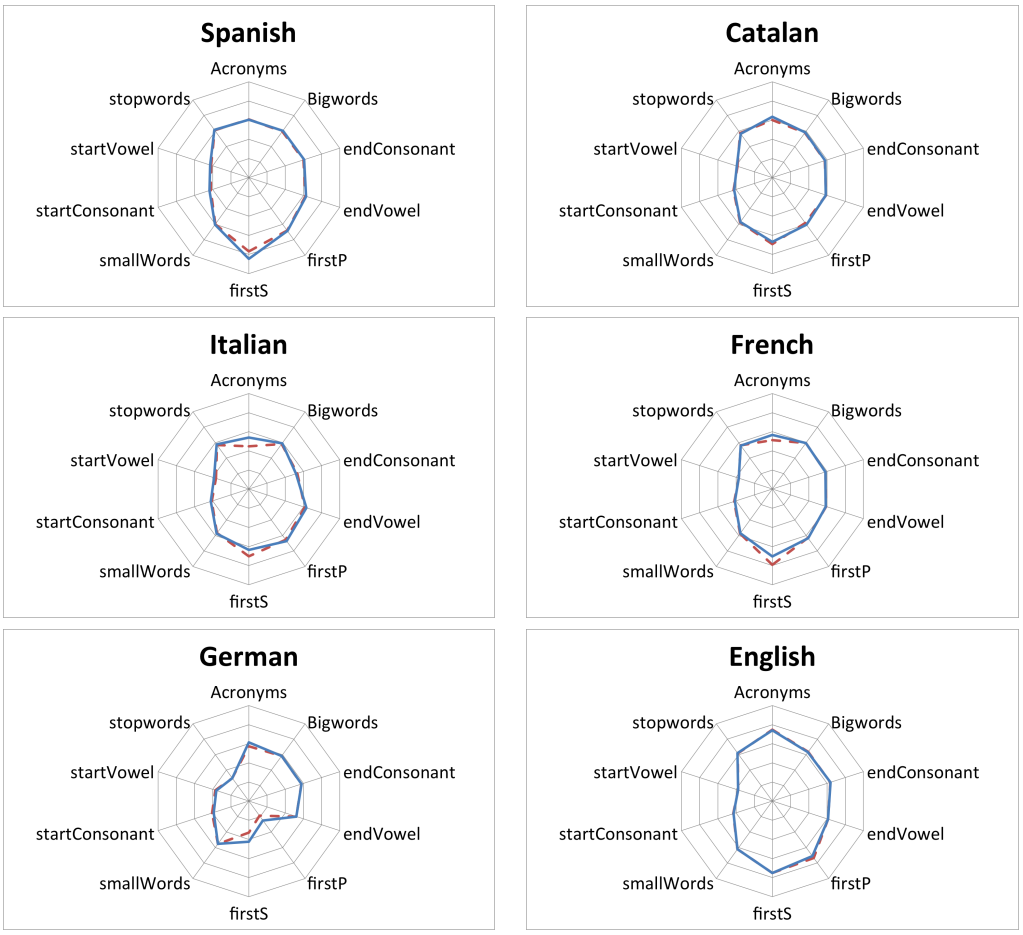


Figure 5.2: Distribution of word-oriented features in the posts of men and women across languages; solid line (male), dotted line (female); where the dotted line does not show, it overlaps with the solid one.

We can also see that the difference between genders is larger in the first figure than in the second one. Punctuation features can be considered highly stylistic features that are used in a subconscious way and as a result, the difference between the values of these features and the reference features is larger than in the case of word-oriented features.

Some interesting language-contrastive observations of the distribution of features can also be extracted. We can see that the distribution of word-oriented features in all Romance languages that we considered in our experiments is rather similar. Since we eliminated the linguistic bias by normalization, we can hypothesize that this similarity is again due to cultural influences.

5.1.3.6 Conclusions

We used a set of language- and content-independent features that were normalized in order to avoid a bias resulting from the idiosyncratic syntactic, punctuation and writing style characteristics of a language. Compared to state-of-the-art proposals in the field, our set of features is very small. Nonetheless, the results are very competitive.

The conclusion that can be drawn from these experiments is that it is feasible to use the same set of features to determine the gender of the authors of texts written in different languages with high accuracy. The setup of the experiments that we carried out and their outcome make us furthermore hypothesize that if a set of language- and content-independent features could profile the writing of authors effectively, it might be possible to detect the native language of an author writing in a foreign language.

5.1.4 Introducing Discourse Features for Gender Identification in Opinion Pieces

In another round of experiments, discourse features were introduced and part-of-speech features and shape measures applied to coordinate/subordinate clauses were added to the syntactic group of features. This is the first time the full set of features described in Chapter 4 is used.

In the context of these experiments, we address the problem of author gender identification of opinion pieces of online versions of newspapers.

5.1.4.1 Experimental Setup

For the experiments, we use Weka's implementation of Random Forests. To obtain more reliable performance figures, we use 10-fold cross validation, such that the outcome of the classification does not depend on which part of the dataset has been used for training and which part for testing.

The chosen dataset is the EnglishDat (see Chapter 4 for more details).

5.1.4.2 Feature Set

As mentioned, for these experiments we draw upon the full set of features described in Chapter 4. That is, in total, six different types of features are used: Character-based (CB), Word-based (WB), Sentence-based (SB), Dictionary-based (all dictionaries), Syntactic (SynF, SynS, SynP) and Discourse-based (DiscF, DiscS). Refer to Chapter 4 for more information about the feature set.

5.1.4.3 Results and their Discussion

We analyze the performance of each group of features individually as well as the whole feature set by computing the accuracy.

Table 5.11 shows the performance of the system in different scenarios. We see that the character-based group of features proves to be very effective. This group analyzes the usage of punctuation marks such as periods, commas, colons or semicolons. The frequency of commas, for example, can be seen as a highly stylistic choice, and the good performance of this group of features tells us that there are clear patterns that differentiate between genders.

Used Features	**Accuracy**
Character-Based	71.23%
Word-Based	69.51%
Sentence-Based	55.74%
Dictionary-Based	54.86%
Syntactic	76.79%
Discourse-Based	64.32%
Full Set	**84.65%**
Stopword Baseline	66.96%

Table 5.11: Performance of the gender identification system using different subsets of features.

We also see that our syntactic group of features is very effective, analyzing the structure of the phrases gives us valuable information that helps the classifier predict the gender of the authors effectively. The performance of the system with the full set of features achieves very competitive results, predicting correctly in more than 84% of the cases.

The performance of the system is compared to a baseline that consists of individual frequencies of stopwords found in the texts (we consider the list of stopwords provided by the NLTK Python Toolkit). It is apparent that our approach outperforms the stopword list classification baseline by a large margin (see (Arun et al., 2009) for an example of gender identification using stopword frequencies).

To analyze what features were the most distinctive in the classification process, we computed the information gain of each feature. The 20 features that were most distinctive are the following:

> • colons, • quotations, • syntactic width, • first person plural pronouns, • usage of commas, • usage of pronouns, • subordinate clause frequency, • standard deviation of word length, • syntactic ramification factor, • usage of stop words, • usage of exclamations, • discursive ramification factor, • discursive depth, • usage of nouns, • usage

of elaborations, • coordinate clause width, • words per sentence, • vocabulary richness, • usage of hyphens, • usage of appositions

Even though the results of the discourse features by themselves were not the highest, we can see that some of the characteristics of the discourse trees are among the features with more information gain. The results also confirm that punctuation marks are very distinctive. Vocabulary richness is very distinctive in the classification process (looking into the feature values we saw that female authors tend to have richer vocabulary).

The high information gain for the frequency of subordinate clause frequency, syntactic width, and syntactic ramification factor tells us that measuring the complexity of the syntactic structures and analyzing the syntactic trees, generates very useful features to differentiate between genders.

5.1.4.4 Conclusions

The full feature set presented in Chapter 4, which is composed mainly of syntactic and discourse features, predicted the gender of the author of texts correctly in more than 84% of the cases, proving its effectiveness once again.

If we compare the results of this experiment with the previous approach (described in Section 5.1.3), which also used the EnglishDat corpus, we can see how the full set of features outperforms the reduced set, improving the accuracy of the system by more than 4% (80.24% vs 84.65%). This improvement proves the relevance of discourse features.

5.1.5 Author Identification in Blog Posts

The first version of our feature set was already successfully applied to gender identification using blog posts in English and in several other languages. Small experiments on age and language identification had also been performed. At that point, we wanted to assess whether our feature set is able to capture the writing style of an author. In the following experiments, the first version of our approach is applied to authorship attribution, a much more challenging task, in which the goal is to distinguish between the writings of 26 different authors. In this case, discourse features and the extended set of syntactic features (which include parts-of-speech frequencies and shape measures for coordinate and subordinate clauses) are not considered.

In what follows, we introduce a set of features mainly of syntactic dependency and structural nature and prove that these features are able to distinguish between 26 different authors with high accuracy. Using several state-of-the-art approaches as baselines, we furthermore show that these features lead to a better performance.

5.1.5.1 Experimental Setup

The selected feature set for this experiment is composed of the character- (CB), word- (WB), sentence- (SB), dictionary- (DBInt, DBPol, DBAbbrev, DBCurse and DBDisc), and syntax-based features (SynF and SynS).

The dataset that was used for our experiments is the AuthorshipDat (see Chapter 4 for more details).

For classification, we use Weka's implementation of Random Forests, using 10-fold cross-validation. In other words, we consider author identification as a multi (26)-class classification problem (each considered author being represented by a class).

To contrast the performance achieved with our features against the performance achieved with some of the features discussed in state-of-the-art literature, we implemented seven different baselines.

For all baselines, we used the same classifier and dataset as used with our model. The first two baselines use normalized frequencies of function words, used in (Zhao and Zobel, 2005). Since the list of function words that was used in the original is not available, we used the lists available at `http://myweb.tiscali.co.uk/wordscape/museum/funcword.html` ("function word list 1" in Table 5.7 below) and `http://www.sequencepublishing.com/academic.html` ("function word list 2").

The next two baselines use normalized frequencies of parts of speech and normalized frequencies of stop words, respectively.[3]. The last three baselines use a combination of the features of the first four baselines.

5.1.5.2 Results and Discussion

Table 5.12 shows the performance of Weka's Random Forests algorithm with our features, compared to the performance with the baseline feature sets.

We can observe that the classifier trained on our features outperforms the baselines by a wide margin. The obvious explanation for this is that the accuracy of an approach that uses function words depends heavily on the choice of the words in the precompiled list. Adding part of speech improves the baseline's accuracy, but structural features still outperform it. To further compare the performance of our features with the performance of FW2, see Figure 5.3. The confusion matrices are very illustrative in that they show where the classifier erred and what the cause for this was.

The function word list approach works reasonably well in cases in which 200 or more texts from one author are available for training. With authors that have

[3]The list of stop words that was used is available in Python's Natural Language Toolkit (NLTK).

Approach	Accuracy
Our Features	**77.65%**
Function word list 1 (FW1)	56.64%
Function word list 2 (FW2)	60.63%
Part of Speech (PoS)	57.27%
Stopwords	59.35%
FW1 + PoS	65.77%
FW2 + PoS	65.92%
Stopwords + PoS	66.29%

Table 5.12: Results with our features, compared with baseline features.

less than 100 texts, the results are much worse. This behavior can be observed in several cases. For example, in the case of class "i", our model predicts correctly 79 instances of the class, while the baseline approach predicts only 9 of them correctly. The situation is similar with the classes "j", "o", "y", "aa", and "v". That is, features that incorporate syntactic phenomena lead to a more accurate author identification. The use of function words is partially also stylistically motivated. But partially their use is purely grammatical (as, e.g., in the case of governed prepositions). Therefore, a larger training dataset is necessary to adequately cover the stylistic use of function words.

5.1.5.3 Conclusions

We have shown that a relatively small set of features composed mainly by syntactic dependency features is very competitive in the author attribution task. The accuracy achieved in a 26-class supervised machine learning experiment outperformed the baselines by a large margin. This is quite promising and could have great impact in the world of plagiarism detection.

5.1.6 Author and Gender Identification using Literary Texts and Blog Posts

At this point, we successfully applied our approach to gender, age, language and author identification, using both texts in English and in other languages. Now, we want to perform author and gender identification using the same approach, but using two types of texts: blog posts and literary texts. This is the first time in which the feature set is applied to two different text genres. The blog post dataset that is used in these experiments, was used in the experiments described in Section 5.1.5, obtaining 77.65% of accuracy using a reduced part of our feature set. In this

a	b	c	d	e	f	g	h	i	j	k	m	n	o	p	q	r	s	t	u	v	w	x	y	z	aa	classified as
179	0	1	3	11	2	0	0	0	0	0	3	2	0	0	0	1	1	0	7	0	1	1	0	0	0	a = author1
0	244	1	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0	0	0	0	0	0	0	1	0	b = author2
0	2	200	0	0	0	0	0	0	0	2	0	2	0	0	0	0	1	0	0	0	0	1	0	0	0	c = author3
1	0	0	182	3	18	0	4	1	0	0	10	1	0	1	0	5	0	1	9	0	4	1	0	0	0	d = author4
14	1	0	1	161	3	0	1	2	0	0	4	1	0	1	0	1	0	0	1	0	2	0	0	0	0	e = author5
5	0	1	19	5	174	0	0	0	0	0	9	0	0	0	1	2	2	0	0	0	3	0	0	0	0	f = author6
1	0	3	0	0	0	154	1	0	1	1	0	0	0	1	0	0	7	1	0	1	0	0	4	1	0	g = author7
0	0	0	0	0	0	1	85	0	0	1	0	1	0	0	0	0	0	0	0	0	2	3	0	0	0	h = author8
0	0	0	8	0	0	0	0	79	0	0	3	2	0	0	0	0	0	0	3	0	11	3	0	0	0	i = author9
0	1	3	0	0	0	7	0	0	65	2	0	1	0	0	0	3	2	0	0	0	0	1	0	2	0	j = author10
0	3	2	1	0	0	0	2	0	0	186	0	4	0	7	1	0	3	0	8	0	2	2	0	27	0	k = author11
4	0	4	22	9	9	0	0	0	0	7	136	2	0	1	0	1	9	6	11	0	10	4	0	0	0	m = author12
1	8	2	0	0	0	0	0	0	0	1	0	204	0	8	0	3	1	0	1	0	0	12	0	6	0	n = author13
0	0	2	1	2	3	0	1	0	0	0	0	9	49	11	0	1	2	0	1	0	3	0	0	4	0	o = author14
1	0	0	0	1	0	0	0	0	0	16	0	10	0	183	0	0	4	0	6	0	2	7	0	17	0	p = author15
2	0	4	1	0	0	0	0	0	0	2	0	1	0	0	95	0	1	1	1	1	0	1	0	0	0	q = author16
0	2	9	1	1	0	1	1	0	1	0	0	4	0	0	0	169	2	0	1	0	0	1	0	1	1	r = author17
0	1	10	0	0	0	5	2	0	0	0	2	1	0	1	0	0	216	2	0	0	0	1	6	1	0	s = author18
4	6	9	4	7	11	1	2	0	3	15	12	9	0	11	1	1	13	52	3	3	4	11	2	8	2	t = author19
2	0	0	4	3	7	0	0	0	0	13	8	1	0	0	0	0	1	2	204	0	0	0	0	1	0	u = author20
0	0	0	0	0	1	0	0	0	0	2	6	2	0	0	4	4	0	1	2	69	0	0	0	0	0	v = author21
1	2	1	1	0	0	0	1	0	0	4	4	3	2	1	0	1	0	0	7	0	198	1	1	0	0	w = author22
1	3	1	0	0	0	0	0	3	0	1	0	12	0	5	0	0	3	1	1	0	2	214	0	1	1	x = author23
2	0	3	1	0	1	4	0	0	0	2	2	1	0	1	1	2	5	1	1	1	0	0	59	0	2	y = author24
1	5	5	2	0	4	1	3	0	0	37	0	4	0	24	1	4	4	0	3	0	2	13	0	137	0	z = author25
1	3	2	0	0	1	1	0	1	0	1	0	5	0	0	0	0	0	1	0	0	1	3	0	2	59	aa = author26

a	b	c	d	e	f	g	h	i	j	k	m	n	o	p	q	r	s	t	u	v	w	x	y	z	aa	classified as
183	0	0	1	6	0	2	0	0	0	0	0	4	0	0	1	2	0	0	3	0	7	3	0	0	0	a = author1
0	243	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	0	1	0	b = author2
0	1	174	1	1	0	0	0	0	0	6	2	1	0	0	0	0	14	4	0	0	1	2	0	1	0	c = author3
3	0	3	160	7	22	6	1	1	0	1	5	4	1	0	1	0	0	0	14	0	7	3	0	2	0	d = author4
31	0	0	7	123	6	0	1	0	0	0	5	1	0	2	0	4	2	2	3	0	4	2	0	0	0	e = author5
7	0	0	30	11	133	1	0	0	0	0	4	5	0	2	0	11	5	0	8	1	3	0	0	0	0	f = author6
3	0	2	3	1	3	137	0	0	0	2	1	1	0	0	1	0	18	0	1	0	0	0	1	2	0	g = author7
3	0	3	0	4	0	0	62	0	0	1	3	0	0	0	0	0	4	0	1	0	8	3	0	1	0	h = author8
2	3	0	20	3	7	0	0	9	0	0	2	20	0	4	0	5	1	3	1	0	6	22	0	1	0	i = author9
3	0	15	4	1	0	1	1	0	21	0	10	0	0	0	4	0	17	1	0	0	2	1	0	6	0	j = author10
0	2	7	0	1	0	0	1	0	0	168	4	1	0	18	0	0	7	2	0	0	4	4	0	29	0	k = author11
8	4	20	31	5	13	7	0	0	0	13	74	5	0	3	1	5	16	6	2	0	12	7	1	2	0	m = author12
2	4	2	0	0	0	0	0	0	0	3	2	186	0	7	0	1	0	2	1	0	4	30	0	3	0	n = author13
4	3	6	2	5	4	0	0	0	0	7	0	5	3	17	3	3	1	1	4	0	11	4	0	6	0	o = author14
3	2	1	0	0	5	0	0	0	0	30	0	4	0	168	0	2	1	0	1	0	1	13	0	16	0	p = author15
2	1	23	1	1	1	0	0	0	1	3	3	1	0	2	43	5	12	0	0	0	3	3	0	5	0	q = author16
0	0	1	2	1	2	0	0	0	0	2	2	3	0	0	0	171	0	2	0	0	3	1	0	5	0	r = author17
3	0	19	2	0	6	10	0	0	0	8	6	4	0	1	1	2	170	2	0	0	6	0	4	4	0	s = author18
5	4	15	6	3	8	1	2	0	0	16	11	20	2	10	0	5	9	29	1	0	15	17	0	15	0	t = author19
8	3	1	14	1	16	3	0	0	0	31	1	4	0	20	0	2	3	1	128	0	1	1	0	8	0	u = author20
4	0	17	6	2	6	13	0	1	1	1	3	1	0	1	2	2	3	0	4	22	2	0	0	0	0	v = author21
3	0	4	0	0	1	1	0	0	0	1	2	16	0	2	1	1	5	1	1	0	183	4	0	2	0	w = author22
0	5	0	0	1	0	0	1	1	0	3	0	25	0	6	0	0	1	0	0	0	3	201	0	2	0	x = author23
1	0	8	7	0	3	12	0	0	0	2	3	0	0	0	1	2	22	0	2	0	1	0	25	0	0	y = author24
5	0	4	0	0	4	0	1	0	0	58	0	9	0	35	0	3	9	2	1	0	3	19	0	97	0	z = author25
7	0	6	2	10	5	0	0	0	0	2	3	13	0	5	1	5	4	2	0	0	5	4	0	3	4	aa = author26

Figure 5.3: Confusion matrices of our model (top matrix) with the FW2 baseline (bottom matrix).

case, we are going to use the full set of features and perform gender identification as well, expanding the previous experiments.

5.1.6.1 Experimental Setup

The selected datasets are the AuthorshipDat and the LiteraryAmerican corpora.

We consider author identification as a multi (16 and 23 respectively)-class classification problem and gender identification as a binary classification problem.

For the classification experiments, we use Weka's implementation of LibSVM (with a linear kernel), using 10-fold cross-validation.

The full set of features described in Chapter 4 is used in the experiments.

To compare the performance achieved with our features against the state-of-the-art literature, we implemented six different baselines and applied them to both classification problems. The first baseline uses normalized frequencies of function words. The next two baselines use normalized frequencies of parts of speech and stop words (The list of stop words that is used is available in Python's Natural Language Toolkit (NLTK)) respectively. The last baselines are a combination of the presented baselines and the majority class classifier. For all baselines, we use the same classifier and dataset as used with our model.

Two experiments have been performed on each dataset. In the first experiment, the goal was to predict which one of the 23 or 16 possible authors wrote each one of the texts of the datasets. The second one used the same feature set and corpora using another labeling of the data. In that case, the goal was to predict the gender of the author of the texts (man vs woman).

5.1.6.2 Results

Let us, in what follows, summarize the results of the experiments.

The results of the first experiment in the AuthorshipDat corpus are shown in Table 5.13. The performance of each individual feature group, the full set, selected feature combinations and the baselines are presented.

The outcome of the first experiment on the LiteraryAmerican dataset is reflected in Table 5.14. The second experiment on the AuthorshipDat corpus resulted in figures displayed in Table 5.15. Table 5.16 shows the outcome of the second experiment on the LiteraryAmerican dataset.

5.1.6.3 Discussion

In what follows, we discuss the results shown in the tables above in both the author identification and gender identification experiments.

In the author identification experiment, Table 5.13 shows that the classifier trained on our features outperforms the baselines when applied to the AuthorshipDat. The obvious explanation for this is that the accuracy of an approach that uses function words depends heavily on the choice of the words in the precompiled list. Adding part-of-speech information improves the baseline's accuracy and outperforms some of our feature combinations. The specific choice of function and stop words is a very powerful means to characterize some of the stylistic choices of the authors and a challenging baseline. Even if in this case the baselines achieve

Features Used	Accuracy
Full Set	**78,16%**
Character-based (CB)	62,18%
Word-based (WB)	51,72%
Sentence-based (SB)	19,84%
Dictionary-based (DB)	27,84%
Syntactic (Syn)	68,44%
Discourse (Disc)	40,11%
Syn + Disc	70,14%
CB + WB + SB + DB	76,42%
CB + WB + SB + DB + Disc	76,17%
Majority Class Baseline	5,95%
Function Word Baseline (FW)	66,99%
Stopword (SW) Baseline	65,81%
Parts of Speech (PoS) Baseline	63,30%
FW + PoS	72,45%
SW + PoS	72,78%

Table 5.13: Results of the author identification experiment on the AuthorshipDat corpus.

competitive accuracies, our full feature set outperforms them, which shows that our feature choice is effective.

Table 5.14 presents the results of the author identification experiment applied to the LiteraryAmerican. In this case, the performance of the feature set achieves a very high accuracy of 91,08%, which outperforms the baselines (that in this case, are also very challenging). Syntactic features perform similarly to the full set of features and outperform the full set when combined with discourse features. The synergy between discourse features and the other types of features is clear: even though the discourse features do not perform well on their own, they improve the accuracy of each feature combination they are part of. The best performing feature combination omits the syntactic features and achieves a very high value of 95.03% accuracy.

To further compare the performance of our features with the performance of the function word baseline in the author identification experiment using the AuthorshipDat, see Figure 5.4 and 5.5. The confusion matrices are very illustrative in that they show where the classifier erred and what the cause for it was.

The function word list approach works reasonably well in cases in which 200 or more texts from one author are available for training. With authors that have less than 100 texts, the results are much worse. This behavior can be observed

Features Used	Accuracy
Full Set	91,08%
Character-based (CB)	76,05%
Word-based (WB)	65,79%
Sentence-based (SB)	29,36%
Dictionary-based (DB)	68,53%
Syntactic (Syn)	91,01%
Discourse (Disc)	64,39%
Syn + Disc	92,99%
CB + WB + SB + DB	93,69%
CB + WB + SB + DB + Disc	**95,03%**
Majority Class Baseline	12,42%
Function Word Baseline (FW)	86,36%
Stopword (SW) Baseline	85,85%
Parts of Speech (PoS) Baseline	81,01%
FW + PoS	86,96%
SW + PoS	87.29%

Table 5.14: Results of the author identification experiment on the LiteraryAmerican dataset.

in several cases. For example, in the case of the class "r", our model predicts correctly 53 instances of the class, while the baseline approach predicts only 7 of them correctly. The situation is similar with the classes "w", "o", "s" and "e". That is, features that incorporate syntactic phenomena lead to a more accurate author identification.

The use of function words is partly also stylistically motivated. But partly their use is purely grammatical (as, e.g., in the case of governed prepositions). Therefore, a larger training dataset is necessary to adequately cover the stylistic use of function words.

In Figure 5.6, we see the cases where the classifier predicts wrongly using our model as well as using the function word baseline on the LiteraryAmerican.

In these confusion matrices, the authors that have smaller amounts of instances are again harder to predict for the function word baseline, compared to our model. This phenomenon can be observed clearly in the case of the writings of Charlotte Perkins Gilman, where 7 instances were correctly predicted by the baseline, compared to the 32 of our system, and in the case of Frances Harper (24 vs 64).

One of the first confusions that was studied was the reciprocal confusion between some of the writings by Willa Cather and Kate Chopin. Both authors were contemporary and had interesting exchanges: Cather wrote an essay to criticize

Features Used	Accuracy
Full Set	**89,97%**
Character-based (CB)	87,91%
Word-based (WB)	81,18%
Sentence-based (SB)	65,01%
Dictionary-based (DB)	71,45%
Syntactic (Syn)	85,17%
Discourse (Disc)	75,34%
Syn + Disc	85,92%
CB + WB + SB + DB	88,09%
CB + WB + SB + DB + Disc	89,22%
Majority Class Baseline	64,11%
Function Word Baseline (FW)	81,72%
Stopword (SW) Baseline	81,46%
Parts of Speech (PoS) Baseline	81,53%
FW + PoS	82,67%
SW + PoS	82,88%

Table 5.15: Results of the gender identification experiment in the AuthorshipDat corpus.

publicly Chopin's novel "The Awakening" and even published "O Pioneers" as a response. Both books shared many similarities and, in general, both authors wrote about sensitive intelligent women who want to be independent (and in many cases fail to do so). Both authors have female characters that try to push female social boundaries and in some cases try to make a living as a man would. It is rather obvious that both authors had an influence on each other, which could explain why they are confused in our confusion matrix.[4]

Nathaniel Hawthorne is confused with Herman Melville by our model. These two authors were directly in contact; several references to one another are found in their writings (Moby Dick was directly dedicated to Hawthorne) and shared homosexual undertones in their writings. Moreover, both authors exchanged many letters in which their affection is clearly shown (some could be interpreted directly as love letters). So the influence between these two authors is also clear.[5].

John Pendleton Kennedy is confused with James Fenimore Cooper. In this

[4]More information about this can be found in `http://realismandnaturalism.blogspot.com.es/2011/10/kate-chopin.html` and (Schneider, 2005)

[5]For more information on the topic, see `http://www.hawthorneinsalem.org/ScholarsForum/MMD2461.html`, `https://www.theguardian.com/books/2011/jan/30/herman-melville-mark-twain-parini` and `http://rictornorton.co.uk/melvill2.htm`

Features Used	Accuracy
Full Set	90,71%
Character-based (CB)	81,02%
Word-based (WB)	78,79%
Sentence-based (SB)	73,88%
Dictionary-based (DB)	84,39%
Syntactic (Syn)	90,76%
Discourse (Disc)	75,22%
Syn + Disc	91,46%
CB + WB + SB + DB	90,95%
CB + WB + SB + DB + Disc	**91.78%**
Majority Class Baseline	52,22%
Function Word Baseline (FW)	52,73%
Stopword (SW) Baseline	52,73%
Parts of Speech (PoS) Baseline	74,84%
FW + PoS	76,81%
SW + PoS	76,87%

Table 5.16: Results of the gender identification experiment in the LiteraryAmerican dataset.

case, a fact that could explain this confusion is the friendship between both and the time spent in the U.S. army by the two authors. Kennedy is known for his contributions to a genre made popular by Cooper, namely historical romances in the early American days. These facts could have influenced the writing style of Kennedy to be similar to Cooper's in some cases.[6]

As far as gender identification is concerned, we can see in Table 5.15 that our model predicts correctly the gender of the authors in 89,97% of the cases. This outperforms each presented baseline. It is very interesting to observe that in this case, the syntactic features by themselves are also able to outperform each baseline, showing that the syntactic structure is very stylistic, and that clear patterns exist that are gender-specific. Character-based features achieve 87.91% of accuracy, which is very close to the performance of the full set. Comma, semi-colon and period usage has been proven in the past to be very stylistic; the performance of this feature group indicates that it is also true in this work.

Table 5.16 shows that the system's performance for gender identification on literary texts is also very high. In this case, it outperforms each baseline by a very

[6]More information on the topic can be found in `http://docsouth.unc.edu/southlit/kennedy/bio.html` and `http://www.knowsouthernhistory.net/Culture/Literature/john_pendleton_kennedy.htm`

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	<-- classified as
178	1	0	0	0	3	1	0	7	0	3	1	0	8	3	1	1	1	0	0	0	0	0	a = a1_male
2	**242**	0	1	0	0	1	0	0	0	0	0	0	0	0	3	0	0	1	0	0	0	0	b = a2_male
21	1	**26**	1	0	9	2	0	6	1	0	1	0	11	0	1	1	0	0	2	0	3	1	c = a3_male
1	0	0	**139**	0	3	1	4	1	24	2	1	0	5	3	4	0	0	0	5	1	0	0	d = a4_male
0	5	0	5	**15**	8	26	5	1	3	6	0	4	1	7	30	0	0	0	4	0	0	0	e = a5_female
10	0	1	5	2	**148**	5	13	4	0	14	2	12	1	1	3	0	0	0	20	0	0	0	f = a6_female
1	3	0	2	0	1	**185**	2	5	3	1	0	2	1	6	30	0	0	0	5	0	0	0	g = a7_male
4	1	0	6	0	13	5	**117**	47	4	12	5	0	0	22	3	0	0	0	8	0	1	0	h = a8_female
5	1	0	0	0	0	3	2	**197**	1	1	3	0	4	21	5	0	0	0	6	0	0	0	i = a9_male
1	0	0	5	0	5	2	2	0	**177**	5	0	0	3	1	3	0	0	0	7	1	0	0	j = a10_female
3	0	1	6	0	17	6	13	2	4	**141**	0	12	5	3	1	0	0	1	5	0	0	1	k = a11_female
5	0	0	3	0	2	0	1	5	2	2	**140**	0	15	0	0	0	0	0	1	0	0	4	l = a12_male
1	1	0	4	0	3	4	0	4	0	1	0	**191**	1	0	2	2	0	0	3	0	0	0	m = a13_male
13	0	0	3	0	6	1	2	9	3	4	13	1	**183**	1	2	0	0	0	6	0	0	3	n = a14_male
2	0	0	7	0	0	7	2	29	1	7	0	2	1	**170**	16	1	0	0	4	0	0	0	o = a15_male
0	4	0	0	1	0	21	0	3	0	2	0	0	0	5	**210**	0	0	1	3	0	0	0	p = a16_male
22	0	0	2	0	5	2	1	1	1	5	0	4	6	2	2	**59**	1	1	8	2	0	0	q = a17_male
2	1	0	10	1	2	16	4	1	3	14	1	2	4	6	1	1	**7**	1	12	0	1	0	r = a18_male
9	2	0	3	0	4	10	5	6	2	5	0	3	4	16	4	3	0	**10**	12	0	1	0	s = a19_female
1	0	0	1	0	7	10	2	6	1	2	0	1	4	0	6	0	0	0	**214**	0	0	0	t = a20_female
10	0	2	2	2	8	1	3	1	4	9	18	1	6	0	0	4	0	0	0	**29**	0	0	u = a21_female
3	0	0	4	0	2	0	4	1	3	0	0	0	4	0	3	0	0	0	8	0	**72**	0	v = a22_male
6	0	0	3	0	15	0	1	4	5	5	12	2	15	0	0	0	0	0	1	0	0	**20**	w = a23_female

Figure 5.4: Confusion matrix of the FWords baseline on the AuthorshipDat corpus.

large margin. The use of function and stop words is ineffective for distinguishing between genders in this case. When part-of-speech information is used by the baselines, their performance improves drastically, showing that in a gender identification experiment, the word category usage is a distinctive characteristic of the writings of men and women. The table also shows that syntactic features are very distinctive, outperforming the full set of features and improving their performance when combined with discourse features.

After successfully analyzing the results of both gender and author identification in literary and blog texts, further insight on the relevance of specific features in each experiment and dataset needs to be drawn. To do so, we computed the information gain of each feature in each one of the presented experiments. Table 5.17 shows the most relevant features per experiment.[7]

[7]To facilitate the understanding of the features that are presented, some clarifications must be made. The upper-cased feature names are either part-of-speech tags or syntactic dependencies and their specific meaning is the following: NNP is the usage of singular proper nouns; CD is the usage of cardinal numbers; POS refers to the usage of words with possessive ending; NN is the usage of singular nouns; IN refers to the usage of prepositions; WP$ is the usage of possesive wh-pronouns; PRP$ refers to the usage of possessive pronouns; NMOD is the usage of nominal

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	<-- classified as
185	2	2	0	0	0	1	0	9	0	0	0	0	4	1	2	2	0	0	0	0	0	0	a = a1_male
1	**240**	1	0	0	0	3	0	0	0	0	0	1	1	0	3	0	0	0	0	0	0	0	b = a2_male
10	0	**55**	0	0	1	0	0	1	1	0	4	0	7	3	0	0	1	0	3	0	1	0	c = a3_male
0	0	0	**151**	1	2	1	0	1	25	4	1	1	0	2	0	0	0	0	5	0	0	0	d = a4_male
1	0	0	0	**76**	9	2	4	1	1	3	1	1	0	3	5	0	0	0	11	0	2	0	e = a5_female
0	0	0	6	7	**184**	1	3	2	1	21	0	3	1	0	0	0	3	0	4	2	2	1	f = a6_female
2	3	0	0	0	0	**207**	2	3	1	1	0	1	4	7	14	0	1	1	0	0	0	0	g = a7_male
0	0	0	3	1	4	0	**212**	11	4	8	0	0	2	2	0	0	0	0	0	1	0	0	h = a8_female
10	1	1	0	0	2	5	5	**204**	0	1	0	0	1	12	1	0	1	0	3	0	2	0	i = a9_male
0	0	0	28	0	3	2	5	0	**158**	2	0	0	3	2	2	1	0	0	3	3	0	0	j = a10_female
0	0	0	9	4	29	0	16	0	3	**148**	0	1	1	2	0	1	2	0	4	1	0	0	k = a11_female
1	0	0	0	2	1	1	0	3	1	0	**141**	1	12	2	0	1	1	0	4	4	3	2	l = a12_male
4	5	0	0	0	1	5	1	1	0	1	1	**187**	1	1	1	0	0	1	2	2	3	0	m = a13_male
13	1	3	5	0	0	0	5	4	1	1	8	3	**176**	7	3	6	1	0	6	0	1	6	n = a14_male
3	2	0	3	0	0	12	7	18	7	0	0	0	5	**178**	11	1	0	0	0	0	2	0	o = a15_male
0	0	0	0	0	0	16	0	6	3	0	0	0	1	4	**215**	1	0	0	3	0	1	0	p = a16_male
5	0	0	0	0	0	1	1	0	0	0	0	0	2	0	0	**114**	0	0	0	1	0	0	q = a17_male
0	0	0	3	0	0	3	2	4	0	9	1	3	1	3	1	0	**53**	0	3	0	4	0	r = a18_male
1	0	0	4	0	3	12	0	6	3	1	0	1	4	8	0	0	0	**52**	4	0	0	0	s = a19_female
0	2	0	4	5	2	4	13	2	4	2	1	2	2	2	3	1	1	0	**204**	0	1	0	t = a20_female
0	0	0	0	0	0	0	1	3	5	0	0	1	0	0	0	15	0	0	0	**75**	0	0	u = a21_female
0	0	0	1	0	0	0	0	1	0	0	0	1	0	1	5	0	0	0	11	0	**84**	0	v = a22_male
2	1	0	0	0	1	1	5	3	3	3	5	5	6	0	0	3	2	0	1	0	1	**47**	w = a23_female

Figure 5.5: Confusion matrix of our model on the AuthorshipDat corpus.

We observe that some of the features are relevant in most experiments (as, e.g., comma and quotation usage, the mean number of characters per word, usage of quotations or the vocabulary richness), while others are specific to some experiments (as, e.g., negative and emotion words). Negative and emotion words are relevant only in the gender identification experiments, which could be related to the different perception of emotions and the way they are described by each gender.

Note that there are also patterns which depend on the data that is being used: first person singular pronoun usage becomes relevant and distinctive when using the AuthorshipDat dataset(while it is not distinctive in the LiteraryAmerican corpus), which indicates that some of the authors express personal opinions in their blog posts while others are more neutral in their discourse.

The tendency for the use of passive voice is a distinctive trait of authors in the LiteraryAmerican corpus, as is the syntactic width, discourse ramification factor, width and depth and the usage of modifier relations. All of the mentioned features

modifiers; PMOD refers to the usage of preposition modifiers; LOC refers to locative adverbial usage; APPO is the usage of appositions; LGS is the usage of logical subjects of a passive verb; PRP refers to adverbial of purpose usage and PRN to parenthetical constructions.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	<-- classified as
189	1	1	0	0	1	0	0	0	0	0	0	0	0	2	1	a = HermanMelville
0	**86**	2	0	0	1	0	0	0	0	0	0	0	1	0	0	b = LouisaMayAlcott
0	1	**119**	0	0	0	0	0	0	0	0	0	0	0	0	0	c = SusanWarner
1	0	0	**80**	2	0	1	0	0	2	0	0	0	0	1	0	d = HenryJames
0	0	0	1	**109**	2	3	0	1	1	1	0	0	0	0	0	e = SusanGlaspell
2	0	2	0	0	**123**	0	0	7	1	0	0	2	0	0	0	f = WillaCather
0	0	1	3	1	1	**67**	0	0	1	0	0	0	0	0	0	g = EdithWharton
0	0	1	0	0	0	0	**64**	2	2	0	0	0	0	0	0	h = FrancesHarper
1	0	0	0	3	10	3	3	**76**	3	2	1	1	0	1	0	i = KateChopin
0	0	6	4	2	1	3	0	1	**84**	0	0	0	0	0	0	j = WilliamDeanHowells
4	2	1	1	0	2	0	0	0	1	**99**	1	1	0	0	0	k = MarkTwain
0	1	0	0	0	0	0	0	0	0	0	**86**	1	6	1	0	l = JohnPendletonKennedy
0	0	1	1	1	0	0	0	1	0	0	0	**44**	0	0	0	m = FrankNorris
0	0	0	0	0	0	0	0	0	0	0	2	0	**98**	0	0	n = JamesFenimoreCooper
7	0	0	0	0	0	0	0	0	0	0	1	0	0	**74**	0	o = NathanielHawthorne
1	0	1	0	2	0	0	0	1	0	0	0	1	0	0	**32**	p = CharlottePerkinsGilman

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	<-- classified as
178	1	1	0	0	1	0	0	1	0	8	2	1	0	2	0	a = HermanMelville
1	**83**	0	0	0	1	0	0	2	2	1	0	0	0	0	0	b = LouisaMayAlcott
4	2	**104**	1	1	2	1	0	1	1	0	0	0	0	3	0	c = SusanWarner
6	0	4	**58**	6	0	5	1	2	5	0	0	0	0	0	0	d = HenryJames
1	0	1	5	**98**	3	0	1	3	4	0	0	0	0	1	1	e = SusanGlaspell
3	2	4	1	6	**99**	3	0	9	4	3	1	0	1	0	1	f = WillaCather
2	1	0	4	4	3	**53**	0	4	2	0	0	0	1	0	0	g = EdithWharton
3	5	7	1	0	6	0	**24**	6	4	6	1	0	3	3	0	h = FrancesHarper
3	0	2	2	8	9	3	1	**60**	7	2	3	2	0	1	1	i = KateChopin
8	1	8	1	4	6	5	2	4	**53**	1	1	3	2	1	1	j = WilliamDeanHowells
14	1	0	0	0	7	0	2	0	2	**83**	3	0	0	0	0	k = MarkTwain
4	0	0	1	0	0	0	0	0	0	0	**89**	0	1	0	0	l = JohnPendletonKennedy
1	0	0	1	3	2	0	0	1	0	0	0	**37**	3	0	0	m = FrankNorris
5	0	2	1	0	0	0	0	2	0	0	1	0	**86**	3	0	n = JamesFenimoreCooper
11	0	1	2	0	1	0	2	0	0	0	6	1	6	**51**	1	o = NathanielHawthorne
2	2	6	1	1	5	6	1	4	1	1	0	0	1	0	**7**	p = CharlottePerkinsGilman

Figure 5.6: Confusion matrices of our model (top matrix) with the FWords baseline (bottom matrix) on the LiteraryAmerican dataset.

characterize the writing complexity and inner structure of the texts. Their relevance indicates that deep linguistic features are a very powerful profiling tool to distinguish between writing styles.

The main conclusion that can be drawn from the most relevant features per experiment is that our feature set is versatile enough to achieve high accuracies in different scenarios, while in each one of these scenarios, different features are most distinctive. The feature set has not been adapted to each specific classification problem and dataset and still facilitates high values of accuracy in each shown

Author Id. LiteraryAmerican	**Author Id. AuthorshipDat**	**Gender Id. LiteraryAmerican**	**Gender Id. AuthorshipDat**
Semicolons	Disc. RamFactor	Periods	Quotations
Commas	Quotations	Words Per Sent STD	NNP
Chars Per Word STD	Disc. Width	Commas	Upper Cases
Periods	Disc. Depth	Chars Per Word STD	Vocab. Richness
Hyphens	Vocab. Richness	Curse Words	FPers Sing Prons
Words Per Sent STD	Uppers	Words Per Sent	Disc. RamFactor
Vocab. Richness	CD	Words Per Sent Range	Disc. Width
Determiners	Past Verbs	Indifference Words	POS
Disc. Depth	Modifier Relations	Indifference Ratio	Disc. Depth
NN	FPers Sing Prons	NMOD	Commas
NMOD	Words Per Sent	IN	CD
Syn. Width	Colon Usage	PMOD	LOC
Disc. RamFactor	Conjunctions	NN	Percentage
Disc. Width	Chars Per Word	Afraid Ratio	Word Range
Present Verbs	NNP	Inspired Words	Negative Words
Modifier Relations	PMOD	APPO	Chars Per Word
LGS	Commas	Elaboration	WP$
PRP$	Word Range	Upper Cases	Condition
Chars Per Word	NNS	Angry Ratio	Past Verbs
Quotations	PRP	PRN	Superlatives

Table 5.17: 20 features with more information gain per scenario.

case, beating competitive baselines.

5.1.6.4 Conclusions

In this set of experiments, the proposed feature set was applied using standard machine learning techniques in two different scenarios: using blog posts and literary texts. In both cases, the accuracy was very competitive, outperforming all the implemented baselines by a large margin and achieving impressive accuracy values. This is promising and could have great impact in the applications oriented towards plagiarism detection, forensic linguistic investigation, literary studies or even marketing studies.

5.1.7 Author, Book, Origin and Gender Identification using Literary Texts

The experiments described in this section are similar to the previous ones. The same set of features and some of the same types of texts are used, but the presented experiments are different. In this section, we focus on literary texts and perform gender, author, origin and book identification. Texts from British and American authors are considered. A thorough process of error analysis both at an author and at a book level is provided.

We also show that with our features, we outperform the best models in the PAN 2014 author verification shared task (Stamatatos et al., 2014) on a literary genre dataset, proving again, the effectiveness of our approach.

5.1.7.1 Experimental Setup

Support Vector Machines (SVMs) with a linear kernel is chosen for classification. The full set of features is used in all the experiments. Let us introduce now the data on which the trained models have been tested.

5.1.7.2 Datasets

We use three datasets in our experiments. The first two are the LiteraryBritish and the LiteraryMerged datasets. The third dataset is publicly available[8] and was used in 2014's PAN author verification task (Stamatatos et al., 2014). It contains groups of literary texts that are written by the same author and a text whose author is unknown (henceforth, "PANLiterary").

5.1.7.3 Experiments, Results and Discussion

We carried out five experiments; the first three of them on the LiteraryDataset, the next one uses the LiteraryMerged and the last one on the PANLiterary dataset. The LiteraryDataset experiments targeted gender identification, author identification, and identification to which of the 54 books a given chapter belongs, respectively. In the LiteraryMerged, gender, author, book and origin (British vs American) identification are performed. The PANLiterary experiment dealt with author verification, analogously to the corresponding PAN 2014 shared task.

Let us now discuss the results of all five experiments.

[8] http://pan.webis.de/clef14/pan14-web/author-identification.html

Used Features	Accuracy Gen	Accuracy Auth
Complete Set	**90.18%**	**88.34%**
Char (C)	67.65%	37.76%
Word (W)	61.79%	38.54%
Sent (S)	60.35%	17.12%
Dict (Dt)	60.62%	17.90%
Discourse (Dc)	69.99%	42.61%
Syntactic (Sy)	88.94%	82.82%
C+W+S+Dt+Dc	80.76%	69.72%
C+W+S+Dt+Sy	89.96%	87.17%
Sy+Dc	89.35%	83.88%
C+W+S+Dt	73.89%	42.55%
MajClassBaseline	53.54%	9.93%
2GramBaseline	79.25%	75.24%
3GramBaseline	75.53%	62.63%
4GramBaseline	72.39%	39.65%
5GramBaseline	65.81%	26.94%

Table 5.18: Results of the gender and author identification experiments.

5.1.7.3.1 Gender Identification

The gender identification experiment is casted as a supervised binary classification problem. Table 5.18 shows in the column 'Accuracy Gen' the performance of the SVM with each feature group separately as well as with the full set and with some feature combinations. The performance of the majority class classifier (MajClassBaseline) and of four different baselines, where the 300 most frequent token n-grams (2–5 grams were considered) are used as classification features, are also shown for comparison.

The n-gram baselines outperform the SVM trained on any individual feature group, except the syntactic features, which means that syntactic features are crucial for the characterization of the writing style of both genders. Using only this group of features, the model obtains an accuracy of 88.94%, which is very close to its performance with the complete feature set. When discourse features are added, the accuracy further increases.

5.1.7.3.2 Author Identification

The second experiment classifies the texts from the LiteraryDataset by their authors. It is a 18-class classification problem, which is considerably more challenging. Table 5.18 (column 'Accuracy Auth') shows the performance of our

model with 10-fold cross-validation when using the full set of features and different feature combinations.

The results of the 10-fold author identification experiment show that syntactic dependency features are also the most effective for the characterization of the writing style of the authors. The model with the full set of features obtains 88.34% accuracy, which outperforms the n-gram baselines. The high accuracy of syntactic dependency features compared to other sets of features proves again that dependency syntax is a very powerful profiling tool that has not been used to its full potential in the field.

In order to obtain further details on where our model fails, we provide the confusion matrix in Figure 5.7.

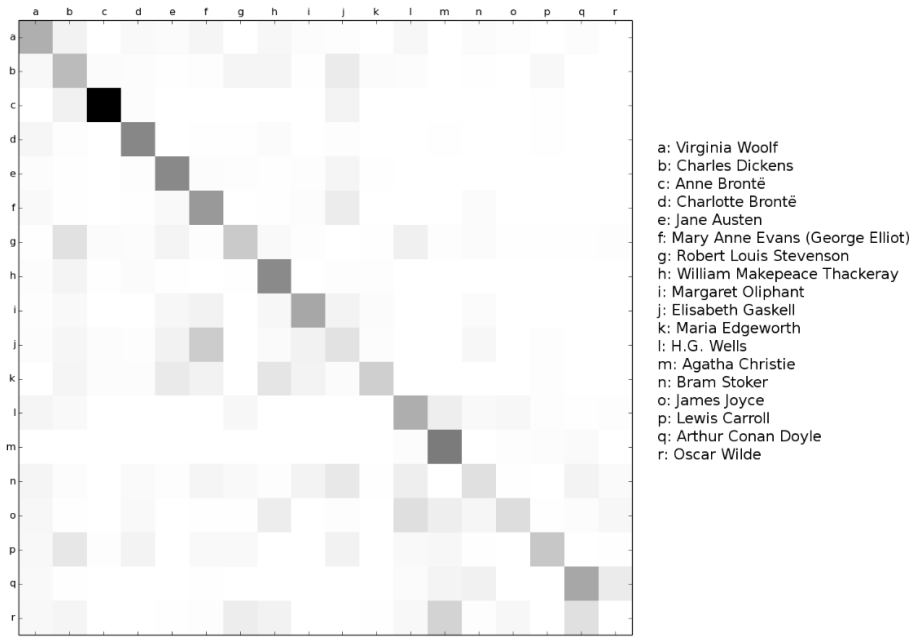


Figure 5.7: Confusion Matrix

We can see that some authors are recognized by the model reliably; see, e.g., Anne Brontë, her sister Charlotte, Jane Austen, Agatha Christie, William Makepeace Thackeray and Arthur Conan Doyle. This is not the case with Oscar Wilde, who is misclassified in the majority of the cases – likely because of the small number of instances we have for this author: the three selected books are split into a very small number of chapters. To correct this issue, another splitting criteria should be followed or additional books should be used.

The novels by Elisabeth Gaskell are confused with the novels by Mary Anne Evans (George Elliot), Jane Austen and Margaret Oliphant. This is likely because

not only do all of these authors share gender, but Austen is also considered to be one of the main influencers of Gaskell. From Margaret Oliphant, it is known that she shared in her prose some of the themes also tackled by Gaskell. In some cases, Oliphant's novels are confused with the prose of Charles Dickens. This might be because both authors share the tendency to describe social tensions, focusing on the least fortunate. Agatha Christie is predicted correctly most of the times. When she is confused with another author, it is with Arthur Conan Doyle. This may not be surprising since both authors share general themes within the crime and mystery genre. Bram Stoker's style is not characterized as accurately. Some of the mispredictions and, in particular, the confusion of Stoker (the author of *Dracula* and other books of the same genre) with Arthur Conan Doyle and H.G. Wells can be explained. Thus, in the novels of both Doyle and Stoker, mystery is one of the main characteristics, and with Wells' novels Stoker's work shares the fiction characteristic. Other mispredictions (such as the confusion of Stoker with Elisabeth Gaskell) require a deeper analysis or signal the need for more training material.

5.1.7.3.3 Source Book Identification

To further prove the profiling potential of syntactic and discourse features, we carried out an additional experiment. The goal was to identify from which of the 54 books a given chapter is, making use of syntactic and discourse features only. Using the same method and 10-fold cross-validation, 83.01% of accuracy was achieved. The interesting part of this experiment is error analysis. *Silas Marner*, written by Mary Anne Evans (known as George Elliot), is one of the books that created the highest confusion; it is often confused with *Mill on the Floss* written by the same author. *Kidnapped* by Robert Louis Stevenson, which is very different from the other considered books by the same author, is confused with *Treasure Island* also by Stevenson, and *Great Expectations* by Charles Dickens. *Pride and Prejudice* by Jane Austen is confused with *Sense and Sensibility* also by her. The majority of confusions are between books by the same author, which proves our point further: syntactic and discourse structures constitute very powerful, underused profiling features (recall that for this experiment, we used only syntactic and discourse features; none of the features was content- or surface-oriented). When the full set of features was used, the accuracy improved to 91.41%. In that case, the main sources of confusion were between *Agnes Grey* and *The Tenant of Wildfell Hall*, both by Anne Brontë and between *Silas Marner* and *Mill on the Floss*, both by G. Elliot.

5.1.7.3.4 LiteraryMerged Experiments

In previous sections we proved that our feature set is very effective identifying the author, gender and book in the LiteraryBritish dataset. In this section, the goal is to see how effective the same feature set is, predicting gender, author, book and origin in a harder problem, where the texts provided to the SVMs are written by 34 different authors from two different geographic origins (British and American). Gender and origin identification are binary classification problems, author identification is casted as a 34-class classification problem and book identification is a 102-class classification problem. SVM is used to classify.

Table 5.19 shows the results of the three classification problems.

Gender	**Author**	**Origin**	**Book**
85,48%	85,34%	82,25%	89,77%

Table 5.19: Results of the experiments performed in the LiteraryMerged corpus.

As we can see, even though the problem is harder, the feature set is versatile enough to distinguish between genders, authors, books and origins effectively.

To look further into the results of the two most challenging tasks of the four (book and author identification), the accuracy that our system obtains per author is shown in Table 5.20 and the accuracy per book is computed and shown in Table 5.21.

In this table, we can see that Oscar Wilde and H.G. Wells are the most confused authors. On the other hand, Anne Brontë, Jane Austen, Henry James and Frances Harper are recognized almost in every case.

To dig deeper into the mistakes of our model, we computed the accuracy per book, to see which were the books that caused our system the most trouble. This information is displayed in Tables 5.21 and 5.22 (table was divided in two parts to fit better).

After analyzing the performance of our system per book, we can see that the easiest books to predict are: *Sylvie and Bruno*, *And then there were none*, *Dubliners*, *Herland*, *Dracula's guest*, *The American*, among others, which are correctly classified every time (14 of the books have 100% precision). On the other hand, the books with worst performance are the following: *House of the seven gables*, *Castle Rackrent*, *The crux*, *Through the looking glass*, *Piazza tales*, *The scarlett letter* and *Silas Marner*.

The bad performance of *Castle Rackrent*, *The crux*, *Piazza tales* and *Through the looking glass* is caused by the books being organized only in small numbers of chapters, not letting the classifier have enough material to learn properly. In these cases, the splitting criteria is what causes the problem. *House of the seven gables*

Author	Accuracy
H.G. Wells	65.7%
Henry James	94.9%
Susan Warner	82.2%
Herman Mellville	88.7%
Virginia Woolf	73.3%
Charles Dickens	81.5%
Anne Brontë	95.0%
Susan Glaspell	85.8%
Charlotte Brontë	90.6%
Willa Cather	83.9%
Mary Anne Evans	84.3%
Robert Louis Stevenson	83.6%
Edith Wharton	83.8%
Frances Harper	93.9%
William Makepeace Thackeray	89.5%
Louisa May Alcott	95.4%
Margaret Oliphant	86.7%
Kate Chopin	77.1%
William Dean Howells	80.4%
Jane Austen	95.6%
Elisabeth Gaskell	76.3%
Nathaniel Hawthorne	79.8%
Maria Edgeworth	91.8%
Bram Stoker	74.1%
Mark Twain	87.0%
James Joyce	85.9%
Agatha Christie	82.5%
Frank Norris	81.6%
John Pendleton Kennedy	86.5%
James Fenimore Cooper	92.3%
Arthur Conan Doyle	86.1%
Oscar Wilde	69.2%
Lewis Carroll	92.1%
Charlotte Perkins Gilman	84.8%

Table 5.20: Accuracy per author in the LiteraryMerged author identification experiment.

Book	Author	Accuracy
Strange case of Mr. Jekyll and Mr. Hyde	Robert Louis Stevenson	96.4%
Horse shoe Robinson	John Pendleton Kennedy	93.2%
Oliver Twist	Charles Dickens	98.1%
Pride and prejudice	Jane Austen	83.6%
Iola Leroy	Frances Harper	91.7%
Great expectations	Charles Dickens	94.7%
House of the seven gables	Nathaniel Hawthorne	63.0%
Sylvie and Bruno	Lewis Carroll	100%
Shirley	Charlotte Brontë	90.0%
World wide world	Susan Warner	98.1%
O Pioneers	Willa Cather	88.5%
Little men	Louisa May Alcott	95.5%
The spy	James Fenimore Cooper	97.2%
Miss Marjoribanks	Margaret Oliphant	96.3%
Ulysses	James Joyce	94.4%
The virginians	William Makepeace Thackeray	87.9%
House of mirth	Edith Wharton	87.1%
Night and day	Virginia Woolf	94.4%
The tenant of Wildfell Hall	Anne Brontë	85.5%
Fidelity	Susan Glaspell	89.2%
Little women	Louisa May Alcott	97.9%
Tales and novels vol. 3	Maria Edgeworth	100%
And then there were none	Agatha Christie	100%
The Invisible man	H. G. Wells	78.3%
My Antonia	Willa Cather	95.7%
Mill on the floss	Mary Anne Evans	77.4%
Lair of the white worm	Bram Stoker	88.9%
Agnes Grey	Anne Brontë	72.7%
Nobody	Susan Warner	98.0%
Castle Rackrent	Maria Edgeworth	50.0%
Dubliners	James Joyce	100%
Deer slayer	James Fenimore Cooper	88.6%
Traveler from altruria romance	William Dean Howells	100%
The glory of the conquered	Susan Glaspell	85.4%
Herland	Charlotte Perkins Gilman	100%
Age of innocence	Edith Wharton	89.5%
Emma	Jane Austen	94.6%
Dracula's guest	Bram Stoker	100%
Secret adversary	Agatha Christie	75.0%
The crux	Charlotte Perkins Gilman	60.0%
The American	Henry James	100%
Wives and daughters	Elizabeth Gaskell	93.5%
Trial and triumph	Frances Harper	81.0%
Blithedale romance	Nathaniel Hawthorne	87.5%
Through the looking glass	Lewis Carroll	72.7%
McTeague	Frank Norris	90.0%
Tale of two cities	Charles Dickens	92.9%
The ambassadors	Henry James	100%
Ethan Frome	Edith Wharton	100%
Rise of Silas Lapham	William Dean Howells	80.0%
Voyage out	Virginia Woolf	88.9%

Table 5.21: Accuracy per book in the LiteraryMerged corpus part 1.

Book	Author	Accuracy
Piazza tales	Herman Melville	50.0%
Villete	Charlotte Brontë	95.1%
The song of the lark	Willa Cather	89.2%
Jane Eyre	Charlotte Brontë	94.7%
North and south	Elizabeth Gaskell	98.1%
Quodlibet	John Pendleton Kennedy	90.5%
Tom Sawyer	Mark Twain	97.1%
The scarlett letter	Nathaniel Hawthorne	64.3%
Sense and sensibilty	Jane Austen	80.4%
The two Marys	Margaret Oliphant	100%
Sign of the four	Arthur Conan Doyle	64.3%
Treasure island	Robert Louis Stevenson	86.1%
Alice's adventures in Wonderland	Lewis Carroll	75.0%
Mysterious affair at styles	Agatha Christie	92.3%
Sowing and reaping	Frances Harper	83.3%
Vanity fair	William Makepeace Thackeray	91.0%
The picture of Dorian Gray	Oscar Wilde	100%
Unicorns	James Joyce	93.5%
Whiteladies	Margaret Oliphant	87.3%
Awakening	Kate Chopin	78.4%
What Diantha did	Charlotte Perkins Gilman	72.7%
Daisy	Susan Warner	95.0%
Confidence man	Herman Melville	95.2%
The octopus	Frank Norris	87.5%
Middle March	Mary Anne Evans	95.2%
Prince and the pauper	Mark Twain	89.7%
Jo's boys	Louisa May Alcott	95.7%
The time machine	H. G. Wells	85.7%
The pit	Frank Norris	90.0%
Cranford	Elizabeth Gaskell	100%
Adventures of Sherlock Holmes	Arthur Conan Doyle	76.9%
Dracula	Bram Stoker	90.0%
Last of the Mohicans	James Fenimore Cooper	93.8%
Bayou folk	Kate Chopin	71.4%
Jacob's room	Virginia Woolf	92.9%
Kidnapped	Robert Louis Stevenson	85.7%
The absentee	Maria Edgeworth	89.5%
The hound of the Baskervilles	Arthur Conan Doyle	100%
Canterville ghost	Oscar Wilde	75.0%
Silas Marner	Mary Anne Evans	73.3%
Huckleberry Finn	Mark Twain	93.5%
The war of the worlds	H.G. Wells	82.8%
A hazard of new fortunes	William Dean Howells	83.3%
The turn of the screw	Arthur Conan Doyle	95.8%
Rob of the bowl	John Pendleton Kennedy	92.9%
At fault	Kate Chopin	81.8%
Moby Dick	Herman Melville	92.6%
Barry Lyndon	William Makepeace Thackeray	89.5%
House of pomegranates	Oscar Wilde	75.0%
The visioning	Susan Glaspell	85.7%

Table 5.22: Accuracy per book in the LiteraryMerged corpus part 2.

is divided into 23 chapters, which is also a small number, but analyzing the confusion matrix, it is noteworthy that this book is often confused with *The scarlett letter*, which makes sense, due to the fact that both books are written by Nathaniel Hawthorne. *Silas Marner* gets confused with *Mill on the floss*, both authored by Mary Anne Evans. Other interesting confusions are *The virginians* with *Vanity fair* (by William Makepeace Thackeray both), *O Pioneers* with *Song of the lark* (both by Willa Cather), and *Sense and Sensibility* with *Pride and Prejudice*, both by Jane Austen. The analysis of these confusions tells us that we are effectively profiling the style of the authors.

5.1.7.3.5 PAN Author Verification

After performing many different experiments on our data, we compared the performance of our system with other approaches. To do so, we used the English literary dataset provided in the PAN 2014 shared task on author verification, with our feature set, and compared our results with the other competitors of the task. The PAN 2014 corpus contains pairs of text instances where one text is written by a specific author and the goal is to determine whether the other instance is also written by the same author. Note that the task of author verification is different from the task of author identification. To apply our model in this context, we compute the feature values for each pair of known-anonymous instances and substract the feature values of the known instance from the features of the anonymous one; the feature values are normalized. As a result, a feature difference vector for each pair is computed. The vector is labeled so as to indicate whether both instances were written by the same author or not.

The task performance measure is computed by multiplying the area under the ROC curve (AUC) and the "c@1" score, which is a metric that takes into account unpredicted instances. In our case, the classifier outputs a prediction for each test instance, such that the c@1 score is equivalent to accuracy. In Table 5.23, the performance of our model, compared to the winner and second ranked of the English literary text section of the shared task (cf. (Modaresi and Gross, 2014) and (Zamani et al., 2014) for details), is shown.

Our model outperforms the task baseline as well as the best performing approach of the shared task and the META-CLASSIFIER (MC), by a large margin. The task baseline is the best-performing language-independent approach of the PAN-2013 shared task. MC is an ensemble of all systems that participated in the task in that it uses for its decision the averaged probability scores of all of them.

Approach	Final Score	AUC	c@1
Our Model	**0.671**	**0.866**	**0.775**
Modaressi & Gross	0.508	0.711	0.715
Zamani et al.	0.476	0.733	0.650
META-CLASSIFIER	0.472	0.732	0.645
BASELINE	0.202	0.453	0.445

Table 5.23: Performance of our model compared to other participants on the "PANLiterary" dataset.

5.1.7.4 Feature Analysis

Table 5.24 displays the 20 features with the highest information gain, ordered top-down (upper being the highest) for the experiments on the LiteraryBritish and PANLiterary datasets.[9] Syntactic features prove again to be relevant in all the experiments.

The table shows that there are features that work well for the majority of the experiments. This includes, e.g., the usage of verb chains (VC), syntactic objects (OBJ), commas, predicative complements of control verbs (OPRD), or adjective modifiers (AMOD). It is interesting to note that the Elaboration discourse relation is distinctive in the first two experiments, while the usage of Contrast relation becomes relevant to gender and book identification. These features are not helpful in the PANLiterary experiment, where discourse patterns were not found in the small dataset. The discourse tree width and the subordinate clause width are distinctive in the author identification experiment, while they are not in the other experiments. This is likely because they can serve as indicators of the structural complexity of a text and thus of the idiosyncrasy of a writing style of an individual – as punctuation marks such as periods and commas, which are typical stylistic features. Discourse markers, words with positive sentiment, first person plural pronouns, Wh-Adverbs and modal verbs are distinctive features in the gender identification experiment. The fact that the usage of positive words is only

[9]The features starting with a capital are discourse relations; 'sentence range' is defined as the difference between the minimum and maximum value of words per sentence. 'STD': standard deviation, 'firstP': first person plural pronouns, 'AMOD': Adjective/adverbial modifier f(requency), 'VC': Verb Chain f, 'PRD': Predicative complement f, 'ADV': General Adverbial f, 'P': Punctuation f, 'MD': Modal Verb f, 'TO': Particle *to* f, 'OPRD': Predicative Complement of raising/control verb f, 'PRT': Particle dependent on the verb f, 'OBJ': Object f, 'PRP': Adverbial of Purpose or Reason f, 'CC': Coordinating Conjunction f, 'RBR': Comparative Adverb f, 'PRP$': Possessive Pronoun f, 'WRB': Wh-Adverb f, 'HMOD': Dependent on the Head of a Hyphenated Word f., 'NNP': Singular proper noun f, 'DT': Determiner f, 'VBZ': 3rd person singular present verb f, 'CONJ': Second conjunct (dependent on conjunction) f, 'PUT': Complement of the verb put f, 'LOC-OPRD': non-atomic dependency that combines a Locative adverbial and a predicative complement of a control verb f.

Author	Gender	Book	PANLiterary
pronouns	AMOD	semicolons	quotations
VC	discourse markers	colons	charsperword
AMOD	pronouns	VB	firstS
commas	firstP	PRP	commas
PRD	VC	MD	hyphens
discourse width	ADV	OBJ	NNP
P	MD	acronyms	subordinate depth
TO	Elaboration	VC	DT
Elaboration	TO	IM	CC
present verbs	OPRD	sentence STD	determiners
subordinate width	PRT	parentheses	PRP
quotations	Contrast	commas	discourse markers
OBJ	PRP	periods	VC
CC	Manner-means	stopwords	VBZ
sentence STD	RBR	OPRD	CONJ
nouns	positive words	AMOD	firstP
OPRD	OBJ	Contrast	PUT
PRP$	WRB	exclamations	LOC-OPRD
HMOD	present verbs	PRP$	coordinate width
periods	sentence range	quotations	adverbs

Table 5.24: 20 features with the highest information gain in the experiments on the LiteraryBritish and the PANLiterary datasets.

relevant in the gender identification experiment could be caused by the differences in the expressiveness/emotiveness of the writings of men and women. Punctuation marks become very distinctive in the book identification experiment, where the usage of colons, semicolons, parentheses, commas, periods, exclamations and quotation marks are among the most relevant features of the experiment. Syntactic shape features are distinctive in the author identification and PANLiterary experiments while not as impactful in the rest of the experiments.

The same analysis is done with the experiments on the LiteraryMerged corpus. The 20 features with more information gain per classification problem are shown in Table 5.25.[10]

If we analyze and compare this table with the previous one, we can see that

[10] 'VOC': usage of vocatives, 'PRD-PRP': non-atomic dependency that combines a predicative complement and an adverbial of purpose or reason f, 'LGS': logical subject of a passive verb f, 'LOC': locative adverbial f, 'VB': verb in base form f, 'CD': cardinal number f, 'IM': infinitive verb f, 'COORD': start of a coordinate clause f, 'APPO': apposition f, 'VBD': verb in past tense f, 'NMOD': modifier of nominal f, 'LOC-MNR': non-atomic dependency that combines a locative adverbial and an adverbial of manner.

Author	Gender	Book	Origin
commas	DT	semicolons	parenthesis
periods	commas	colons	VOC
colons	determiners	commas	PRD-PRP
past verbs	PRP	sentence STD	periods
hyphens	abbreviations	acronyms	LGS
VB	LOC	hyphens	DT
Contrast	three char words	vocabulary richness	subordinate width
VC	OPRD	periods	colons
AMOD	CD	syntactic width	hyphens
IM	IM	words per sent	subordinate ramFactor
PRP$	discourse depth	chars per word STD	COORD
COORD	periods	VB	VC
PRD-PRP	past verbs	PRP$	PRP
CONJ	discourse width	parenthesis	determiners
three char words	VB	two char words	discourse width
quotations	APPO	CC	NNP
sentence STD	quotations	COORD	discourse ramFactor
PRP	positive words	CONJ	PRP$
Elaboration	VBD	PRP	commas
syntactic depth	LOC-MNR	pronouns	NMOD

Table 5.25: 20 features with more information gain in the experiments on the LiteraryMerged corpus.

there are several similarities: the usage of positive words is relevant in the gender identification problem as in the previous experiments, the Elaboration discourse relation proves to be distinctive in the author identification experiment as well, and the standard deviation in words per sentence is also a characteristic that helps to distinguish the authors effectively . On the other hand, past verb usage is distinctive in both gender and author identification, while in the previous results, present verbs were more relevant, punctuation marks are much more distinctive in both gender and author identification than in the previous case, and features such as syntactic depth, three character words and usage of dependencies such as LOC COORD and APPO gain relevance. Parentheses usage, subordinate width, discourse width, comma usage or the subordinate ramification factor, among other stylistic characteristics, are some of the features that are relevant in the origin classification problem. In both book and author identification, the punctuation marks are among the most distinctive features. The usage of coordinate clauses is also a common characteristic that is distinctive in both cases. While in author identification syntactic depth is relevant, in book identification, the syntactic width is

what helps the classifier. Comparing the book identification information gains in the LiteraryMerged corpus with the LiteraryBritish dataset, we can see that stopwords are not that relevant in the LiteraryMerged, while vocabulary richness is. The standard deviation in words per sentence and the relevance of the punctuation marks is similar in both cases.

5.1.7.5 Conclusions

We have shown that syntactic dependency and discourse features play a significant role in the task of gender and author identification and author verification. We have also applied our model to perform origin and book identification, proving that our approach is versatile and that it can be applied effectively to many different tasks. With more than 88% of accuracy in both gender and author identification within the literary genre, our models are able to beat competitive baselines.

5.2 Semi-Supervised and Unsupervised Approaches

Even though the majority of the experiments carried out in the context of this thesis use supervised learning, semi-supervised and unsupervised learning have also been considered. The tendency of having scarce labeled data in real-world applications motivates these approaches. In the case of semi-supervised learning, we implement an algorithm that uses a small seed of correctly labeled data with larger quantities of unlabeled data to complement it, proving that unlabeled data helps boosting the performance of the system.

In the case of unsupervised learning, an improved version of K-means is presented. Our approach automatically estimates the number of clusters to form, to then proceed with the clustering. The improvements over the original algorithm as well as the comparison with other standard clustering algorithms is shown below.

5.2.1 A Semi-Supervised Approach for Gender Identification

In the vast majority of the existing works, author profiling and author gender identification are approached as supervised machine learning problems. Supervised learning requires a sufficiently large corpus of clean, correctly annotated training data. However, in many author profiling tasks, such data is not available. Consider, for instance, forensic applications, where only a limited number of writings of the same author can be counted on, or literature studies, where the amount of written material might be sufficient, but not annotated. In this context, semi-supervised learning (or even unsupervised learning) suggests itself as an alternative. The goal of Semi-Supervised learning is to use unlabeled data and a small

sample of labeled data to learn.

Even though there have been Semi-Supervised Learning approaches in lots of related fields, to the best of our knowledge, it has not been used yet in the context of author profiling.

In what follows, we present a modified version of the K Nearest Neighbors (kNN) algorithm as a semi-supervised learning approach that benefits from the usage of unlabeled data to boost the performance of gender identification. We first introduce experimental setup and the semi-supervised learning algorithm that was implemented. Then it is shown how unlabeled data can help boost the performance of gender identification when data might be scarce and demonstrate that the selected features are indeed effective for this task.

5.2.1.1 Experimental Setup

The chosen dataset is the SmallEngDat. In the performed experiments 113 texts were used as the initial training set (with known annotations), 113 texts as test set and the rest of the dataset as unannotated data.

The selected groups of features are the character-based (CB), word-based (WB), sentence-based (SB), dictionary-based (DBDisc, DBInt, DBPol, DBMood, DBCurse and DBAbbrev) and syntactic (dependency frequencies and width/depth of the syntactic trees).

5.2.1.2 Enriched KNN algorithm

Our semi-supervised learning algorithm for gender identification is a modified version of the classic *K nearest neighbors* (kNN) classifier. Given a test instance, this algorithm, identifies the k instances that are closest (in accordance with a vector distance metric such as cosine or Euclidean distance) to the test instance. The test instance is labeled with the most common label among its k neighbors.

Our algorithm works in two phases. In both phases, the feature values are normalized between 0 and 1. Prior to the classification of an instance, both the test instance and the training set instances are normalized by dividing each feature value by its maximum feature value among all involved instances (training set and the instance that is being classified). Using this strategy makes the computed distances meaningful in the vector space that is being used.

The distance is also scaled between 0 and 1. To do so, the Euclidean distance between two instances is divided by the number of features. The reasoning behind this is that, since all the features are scaled between 0 and 1, the maximum value that the Euclidean distance can achieve is the number of dimensions of the vectors. Dividing this value between the number of features scales the distance between the same boundaries and as a result, the scores are also scaled in the same way.

The first phase of the algorithm is the *enrichment phase*; cf. Algorithm 1. The goal of this phase is to expand the initial dataset by giving the unlabeled instances a score for each possible label, ensuring at the same time that these scores are lower than the ones that the labeled instances have (the labeled instance score is the upper bound of the unlabeled scores). Given an unlabeled instance, we get the k nearest neighbors which will be the k labeled instances that have the least Euclidean distance between them and the given test instance. The unlabeled instances that have a score are not considered as possible "neighbors", since this strategy can lead to a lot more noise in the enriched dataset (since the decisions that are made depend on low-reliability instances).

Algorithm 1 Enrichment phase.

```
for u in unlabeled_set do
    kneighbors = getNearestNeighbors(u, train_set, K)
    scores = dict()
    for n in kneighbors do
        scores[n.label] += (n.score[n.label] - n.distance)/K
    end for
    u.setScores(scores)
    train_set.add(u)
end for
```

For each neighbor, we increment the score for the neighbors' correct label by the difference between the score of the neighbors (which will be 1.0 since these are correctly labeled instances) and their Euclidean distance (which, as it was stated before, it is scaled between 0 and 1), divided by the total number of neighbors.

After setting the computed scores and adding the new instance to the training set, the labeled instances will have better scores than the unlabeled ones. By default, every instance that is manually labeled will have a score of 1.0 for their correct label and 0.0 for the incorrect one. The scores represent the probability that an instance has a label. This is a way to make the unlabeled instances useful while prioritizing the correctly labeled ones. This process of assigning probability-based labels can help classification processes in which the manually labeled data is scarce.

The second phase of the algorithm is the *classification phase*; cf. Algorithm 2.

To classify the test instances, first of all, the k nearest neighbors are retrieved the same way as it was done during the *enrichment phase*. Then, for each neighbor, the probabilities for each possible class are added. The class with a better accumulated score provides the label for the test instance. The impact of a manually labeled instance in the neighborhood of a test instance will always be higher

Algorithm 2 Classification phase.

```
for t in test_set do
    kneighbors = getNearestNeighbors(t, train_set, K)
    scores = dict()
    for n in kneighbors do
        for label in n.score.keys() do
            scores[label] += n.score[label]
        end for
    end for
    t.label = getMaxLabel(scores)
end for
```

than the impact of the instances that were added in the first phase.

5.2.1.3 Results and Discussion

To evaluate the effectiveness of our algorithm and the chosen feature set, we designed two experiments. In both, 10% of the dataset was used as training set, another 10% as test set and the rest as unlabeled instances.

To test the behavior of the feature set, we executed only the *classification phase* (as outlined in Algorithm 2), using 10% of the dataset for training and another 10% as test set. Both sets contained the same number of instances per class. To evaluate the accuracy, the classification was executed 1000 times, changing randomly the training and test set in each execution. Table 5.26 displays the accuracy of the classifier for different ks, comparing it to three baselines that follow the " bag of words" approach and that consist in using the frequencies of the 300, 400 and 500 most common words in the training set for classification.

K	Accuracy	BoW300	BoW400	BoW500
12	**74.19%**	66.81%	66.61%	64.39%
22	**72.80%**	66.88%	65.67%	62.74%
27	**71.06%**	64.77%	63.20%	59.49%
34	**69.51%**	65.89%	63.49%	59.56%
45	**69.47%**	62.77%	59.61%	56.10%
67	**65.39%**	59.65%	56.34%	54.32%

Table 5.26: Accuracy of the classification phase.

We can see that even when our classifier has only 113 instances to train and the same amount to test, the performance is quite competitive. Reaching more

than 70% of accuracy in this conditions is a good indicator that the chosen feature set is effective in distinguishing between genders.

To have a better understanding of the performance of our feature set and to see which features distinguish better between genders, the information gain coefficients of the features have been computed. The twenty most distinctive features are the following:

- Vocab. Richness
- Interjections
- HYPH
- TMP
- 2-character words
- Upper cased chars
- Word STD
- Quotations
- Negative Words
- Dot frequency
- Chars. per Word
- First Person Singular Pronouns
- Semicolons
- Acronyms
- Tree Width
- VC
- NMOD
- LOC
- Abbreviations
- HMOD

The features that are upper cased are frequencies of Syntactic Dependencies. Some conclusions can be drawn upon the list with the most distinctive features. First of all, it can be observed that the syntactic features are very relevant: several dependencies are very distinctive. The width of the syntactic trees is also relevant. This measure can be seen as an indicator than the complexity of the discourse between genders differs. At a lower level, this statement holds: the vocabulary

richness and the number of characters per word are also relevant in the classification process.

It is also interesting to note that the percentage of negative words is also relevant. This can be related to the fact that in general, men tend to be less emotionally involved in the stories they write than women. The differences in usage of first person pronouns is also noticeable, could be related to the tendencies of each gender to write about themselves opposed to write about the people around them.

The second experiment measured the accuracy improvement that is obtained by executing both phases. First, the *classification* was executed, as it was done in the previous experiment. After that, the *enrichment phase* was executed and finally, the *classification* was run again with the enriched dataset (this process was also carried out 1000 times, randomizing train and test set each time). Note that the accuracies does not match the ones shown in Table 5.26 due to the random samplings. Table 5.27 shows the improvements in accuracy achieved.

K	**Initial Accuracy**	**Enriched Accuracy**
12	74.01%	76.82%
22	71.11%	74.32%
27	71.56%	73.28%
34	69.14%	72.69%
45	68.33%	71.88%
67	64.22%	69.89%
80	60.01%	68.04%
100	53.99%	66.96%

Table 5.27: Accuracy of the combined classification and enrichment phases.

We observe that our classification algorithm achieves good accuracy already with a small sample of instances for training. We believe that this is due to the composition of the features we use. However, adding more instances in a semi-supervised fashion lets the classification further improve. More precisely, by adding 863 unlabeled instances, our algorithm improves for every k. Note that in the case of a considerably higher number of unlabeled data and the same number of labeled data, an instance selection process would be required to avoid introducing noise to the training set.

A simple instance selection could consist in the analysis of the standard deviation of the scores for each class of the unlabeled instances; the instances with a higher standard deviation in their score than a threshold would be added, the others would be discarded. A clear example of this statement would be to have an instance with probabilities: p(male)=0.55, p(female)=0.45 and an instance with

p(male)=0.2, p(female)=0.8. The second case is clearly a more useful instance. In small datasets, this might not be that problematic, but with thousands of instances, the instances with probabilities like the first case would introduce noise that could make the accuracy of the enriched classification process decrease.

It can be also observed that the value of k is very significant for the classification. Higher values of K give the algorithm more information, but it can also make the problem more susceptible to noise and overfitting (and also increases the computational cost). The higher the number of analyzed neighbors, the higher the chance of finding that one or more of the neighbors are actually outliers or unlabeled instances labeled with low reliability. Lower values of K, make the problem perform better but in bigger datasets, these values should be scaled accordingly, otherwise we would be analyzing local neighborhoods that might too small to be representative.

5.2.1.4 Conclusions

We presented a semi-supervised approach to gender identification that achieved very competitive accuracy on scarce training data using our feature set. An analysis of which features of the presented feature set were most distinctive has been provided. We proved that the enrichment process improved the classification process by probabilistically labeling unlabeled instances analyzing their neighborhood.

5.2.2 Applying Density-Based K-means to Author Profiling

As it was stated before, author profiling and identification are mainly casted as supervised learning problems. Unsupervised (and semi-supervised, as we introduced before) learning strategies can be a viable alternative. Using unsupervised learning, instances that are stylistically similar can be clustered together, without that their labels are necessarily known.

Clustering is helpful even if larger volumes of training material are available, since it can be used to identify correlations between different features and thus to obtain a better view on which feature combination captures best the writing style of a class of authors. But not all unsupervised models serve equally well this task either. Some of the clustering algorithms that are widely known and are used in many different fields (such as, e.g., the standard K-means or Hierarchical/Agglomerative Clustering) can be very restrictive in the sense that they require a precise parametrization and thus presuppose detailed knowledge of the nature of the data.

In what follows, we present an extension of the classic K-means clustering algorithm that needs little parametrization to perform well. The algorithm first

analyzes the provided feature vector space with respect to its instance density distribution and calculates for each identified dense zone a K-means initial centroid. Based on these preprocessing calculations, K-means clusters then the dataset. The number of clusters corresponds to the number of dense zones identified before.

To assess the performance of our *density-based K-means* algorithm, we apply it to the tasks of gender and authorship identification. In addition, we use publicly available data used in different fields to prove that it performs well not only in the field of author profiling, but also in unrelated fields. The main difference between this algorithm and the one presented in Section 5.2.1.2 is that it is completely unsupervised. In this case, every instance is unlabeled and the goal of the algorithm is two-fold: 1) estimate the number of clusters to form, 2) group instances into the number of clusters estimated beforehand.

5.2.2.1 Density-Based K-means

The data topology of our problem does not provide immediate cluster candidates since the classes are not easily separable. This makes this problem non-trivial for a clustering algorithm. Many clustering algorithms require as input parameter the number of clusters into which the data is to be divided. However, it depends entirely on the specific problem whether an *a priori* fixed number of clusters is appropriate or not.

For instance, in the case of clustering texts written by different authors, the texts can be grouped using different criteria (author name, gender, age, academic background, native language, genre, etc). Each of these criteria may lead to a different number of optimal clusters. Also, if we do not limit the number of clusters *a priori*, we might find synergies between different ways of labeling a text (thus, female teenagers from Germany might have in English a similar style as French male teenagers). In what follows, we propose a versatile algorithm that automatically estimates an optimum number of clusters given a feature space for the representation of the data.

The algorithm is divided in two phases: Density Estimation and Clustering. The goal of the first phase is to discover how many dense zones, i.e., zones with an elevated number of data instances, are encountered in the feature space of the data and to determine the centroids of these dense zones. The determined number of dense zones and their centroids serve as input to the second phase that consists of a K-means implementation to carry out the actual clustering: the number of zones is the number of clusters to be formed and their centroids are the initial centroids from which K-means starts. With this two-phase strategy, we mitigate the problem of the classic K-means that its performance is heavily influenced by a poor initialization. It also makes K-means converge in less iterations.

In both phases, the chosen distance metric is the cosine distance. The data is

normalized by dividing the value of each feature by the maximum value that this feature has in the dataset.

5.2.2.2 Density Estimation

The *Density Estimation* phase of the algorithm is outlined in Algorithm 3. In this phase, first, the “Density Center" centroid is computed. Each component of the Density Center centroid is the median of the feature values of each instance. This centroid can be viewed as the center point of the instances of the data. It is used to compute a distance threshold. We will refer to a group of instances clustered together as a “zone”. The $currentZone$ refers to the zone where the instance we are iterating over is located, an instance is the vectorial representation of the feature set of a text of the dataset and $zones$ is the structure where every zone is stored.

Algorithm 3 Density estimation phase outline.

$densityCenter$ = computeDensityCenter(data);
$zones$ = minMerge(data);
T_{DC} = getThresholdDistances($densityCenter$, $zones$);
T_{min} = getThresholdElements($zones$);
$zones$ = distanceMerge($thresholdDistances$, $zones$);
$zones$ = elementMerge($thresholdElements$, $zones$);

In Algorithm 4, the first grouping of the instances into “zones" is carried out: each instance is assigned the same zone as the instance that is closest to it. To do so, we loop over each instance. If the instance is already in a zone (i.e., it has been grouped together with other instances in previous iterations), the current zone is set to this zone. If not, a new zone with only this instance is created and defined to be the current zone. Next, the instance that is closest to the given one is retrieved. If this instance is already in a zone, both zones are merged. If not, the retrieved instance is added to the current zone. If neither of the instances was in any existing zone at the start of the loop, the current zone (which will have the instance we are iterating over and its closest instance) is added to the “zones” structure. The goal of this first grouping stage is to form initial clusters of the instances that are closest.

The output of the first grouping of data instances into zones on a sample gender annotated dataset is illustrated in Figure 5.8. The blue and pink dots are Principal Component Analysis (PCA) projections of instances of men (blue) and women (pink). The larger yellow circles are zone centroids (which are computed by calculating the median of each feature of the elements of each zone); the red circle is the Density Center.

Algorithm 4 First grouping of data instances into zones.

```
function MINMERGE(data)
    zones = list();
    currentZone = list();
    for instance1 in data do
        found1 , found2 = boolean();
        if instance1 in zones then
            currentZone = zones[instance1];
            found1 = true;
        else
            currentZone.add(instance1);
            found1 = false;
        end if
        closestInstance = getClosestInstance(instance1);
        if closestInstance in zones then
            merge(currentZone, zones[closestInstance]);
            found2 = true;
        else
            currentZone.add(closestInstance);
            found2 = false;
        end if
        if not found1 and not found2 then
            zones.add(currentZone)
        end if
    end for
     return zones;
end function
```

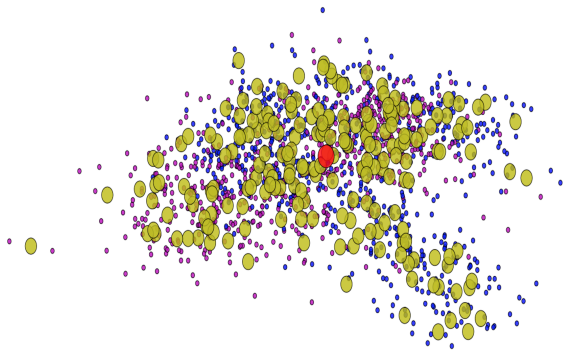


Figure 5.8: PCA projection after the first grouping step.

It can be observed that the first grouping leaves us with a high number of small zones that unite instances that are spatially close. We need thus to limit the number of zones by merging them. To do so, we use two thresholds in sequence: a distance based threshold (T_{DC}) that uses the Density Center and a threshold (T_{min}) that is the minimum number of instances that a zone must have.

T_{DC} is the median value of the distances between the Density Center and the centroid of each of the zones that the first grouping function returned. T_{DC} is our reference distance. If the distance between two zones is smaller than T_{DC}, they are merged into one. If a zone has more than one merging option (more than one zone is at a distance lower than T_{DC}), the closest one is chosen.

T_{min} is a percentage of the total number of instances. In our experiments, we set T_{min} to 1%. The effects of this threshold on the number of zones and accuracy of the approach are analyzed below.

Algorithm 5 shows the merge of the preliminary zones obtained during the first grouping using T_{DC}: we loop until there are no changes in the zones structure. In each iteration, we get the two closest zones (the zones for which the centroids have less cosine distance) and the distance between them. If this distance is lower than the computed T_{DC}, the algorithm merges them.

Algorithm 5 Merge of preliminary zones using T_{DC} as threshold.

```
function DISTANCEMERGE(T_DC , zones)
    changed = boolean(true);
    while changed do
        oldZones = copy(zones);
        zone1,zone2,distance = getClosestZones(zones);
        if distance ≤ T_DC then
            merge(zone1,zone2);
        end if
        changed = hasChanged(oldZones,zones);
    end while
    return zones;
end function
```

Figure 5.9 illustrates the output of Algorithm 5. If we compare this figure to Figure 5.8, it is obvious that the number of zones has been reduced. It is still a very large number that needs to be further minimized. To achieve this, we use T_{min}; cf. Algorithm 6. In this algorithm, we iterate until no changes are made to the zones structure. For each zone, we check if its length (the number of instances it contains) is less or equal than the computed threshold. If this is the case, we look for its closest zone, and merge the two. It could be argued that this merge could lead to noise due to potential outliers, but in this step the centroids are computed

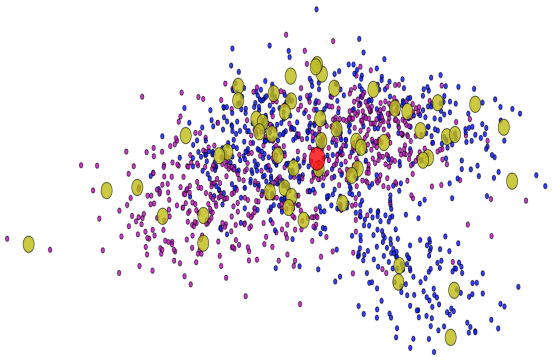


Figure 5.9: PCA projection after the distance merge step.

using the median instead of the mean, so it is less prone to be affected by outliers. The output of this algorithm is presented in Figure 5.10.

Algorithm 6 Element merge.

```
function ELEMENTMERGE(T_min , zones)
    changed = boolean(true);
    while changed do
        oldZones = copy(zones);
        for zone in zones do
            if zone.length ≤ T_min then
                closestZone = getClosestZone(zone);
                merge(zone,closestZone);
            end if
        end for
        changed = hasChanged(oldZones,zones);
    end while
    return zones;
end function
```

As we can see, the number of zones has been reduced drastically (in the image, there are three zones, one is behind the density center). The output of the phase of Density Estimation are the final zones and their centroids. Both serve as input to the next phase of Clustering.

5.2.2.3 Clustering

After computing an estimation of the number of clusters based on the density of the feature space and the centroids of each of the dense zones, these results are

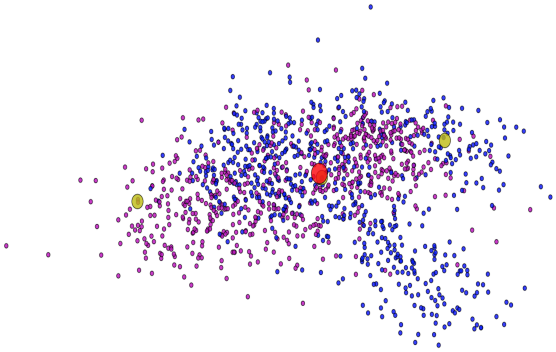


Figure 5.10: PCA projection after the element merge step.

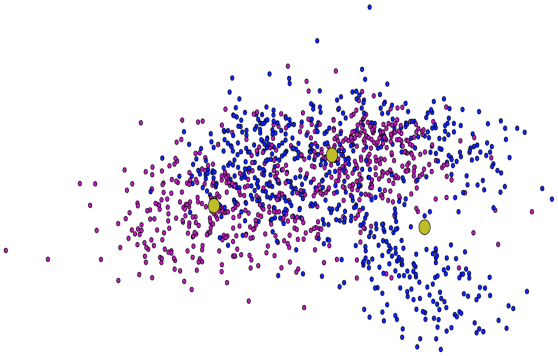


Figure 5.11: PCA projection after K-means.

used to initialize a K-means algorithm. As already mentioned, the number of dense zones is the number of clusters to form and their centroids are the initial centroids with which K-means starts to work. In this phase of our overall algorithm, the centroid calculation is performed by computing the mean (instead of the median, which was used in the density estimation phase) of each component of each instance for each cluster. The chosen distance metric is again the cosine distance.

The output that is produced after the completion of both phases is presented in Figure 5.11. We observe that the centroids have moved, assuming more optimal positions.

5.2.2.4 Experimental Setup

The selected dataset for the gender and authorship experiments is the SmallEngDat labeled either by the gender or the name of the author of a text. Using this

dataset, two experiments are performed: gender and author identification.

For the third experiment, we used publicly available data from the UCI Machine Learning Repository (Lichman, 2013), used in other classification experiments:

- the Spambase Dataset[11] (henceforth, 'Spam'),
- the Image Segmentation Dataset[12] (henceforth, 'Image'),
- the Phishing Websites Dataset[13] (henceforth, 'Phishing'),
- the Banknote Authentication Dataset[14] (henceforth, 'Bank'),
- the Cardiotocography Dataset[15] (henceforth, 'Cardio3' or 'Cardio10', depending on what kind of labeling is chosen. This dataset is labeled for two different problems, 'cardio3' has three possible labels and 'cardio10' has ten) and
- the Blood Transfusion Service Center Dataset[16] (henceforth, 'Blood') (Yeh et al., 2009).

The selected groups of features are the character-based (CB), word-based (WB), sentence-based (SB), dictionary-based (DBDisc, DBInt, DBPol, DBMood, DBCurse and DBAbbrev) and syntactic (dependency frequencies and width/depth of the syntactic trees).

5.2.2.5 Experiments and their Results

To evaluate the performance of the proposed algorithm, we measure the accuracy of the algorithm by adding up the number of instances of the dominant class (the class which has the majority of the instances of the cluster) of each cluster and dividing the sum by the total number of instances in all clusters.

We use four different baselines:

1. Randomly initialized K-means.
2. K-means++ initialized K-means.

[11] http://archive.ics.uci.edu/ml/datasets/Spambase
[12] http://archive.ics.uci.edu/ml/datasets/Image+Segmentation
[13] http://archive.ics.uci.edu/ml/datasets/Phishing+Websites
[14] http://archive.ics.uci.edu/ml/datasets/banknote+authentication
[15] http://archive.ics.uci.edu/ml/datasets/Cardiotocography
[16] http://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center

3. DBSCAN.

4. Agglomerative Clustering. (Ward Jr, 1963).

For 1. and 2., see (Arthur and Vassilvitskii, 2007); 3. and 4. are part of the Python toolkit, “scikit-learn" (Pedregosa et al., 2011).

Both K-means versions as well as Agglomerative Clustering are provided with the correct number of classes to form as input parameters. All baselines are run on the Full feature set.

In the Tables 5.29, 5.31, and 5.32 below, ‘ARK’ stands for the mean accuracy of a randomly initialized (chosing the *n* classes initial centroids at random) 1000 times K-means, ‘AKPP’ for the accuracy of a K-means initialized using the K-means++ strategy, ‘DBSCAN’ for the DBSCAN clustering algorithm, and ‘AggC’ for Agglomerative Clustering.

5.2.2.6 Gender Identification

Table 5.28 shows the performance of the algorithm on different subsets of the presented feature set, with 1% of the total number of instances as the T_{min} threshold introduced above; both the accuracy and the number of zones estimated by the Density Estimation Phase are shown.

Feats	#Zones	Accuracy
Full Set	3	66.75%
Character-Based	6	**69.13**%
Word-Based	10	64.24%
Sentence-Based	9	60.27%
Dictionary-Based	7	63.05%
Syntactic	4	61.70%
Char Word Dict	3	61.58%
Char Dict	5	67.49%
Char Syntax	3	60.94%
Char Dict Syntax	3	59.82%

Table 5.28: Performance of gender identification on SmallEngDat using 1% as T_{min} threshold.

In Table 5.29, the number of zones and the performance of our algorithm with 0.5% of the total number of instances as T_{min} and the same combinations of features as with 1% as T_{min} are shown and contrasted with the performance of the baselines. We can observe an increase of the number of estimated zones, due to the lower threshold as well as a higher accuracy in the majority of the cases.

Feats	#Zones	Accuracy
Full Set	8	77.95%
Character-Based	9	69.94%
Word-Based	14	65.25%
Sentence-Based	10	60.24%
Dictionary-Based	9	61.95%
Syntactic	9	73.33%
Char Word Dict	10	75.68%
Char Dict	8	68.85%
Char Syntax	10	78.08%
Char Dict Syntax	13	**80.11**%
ARK		58.12%
AKPP		57.99%
DBSCAN		58.41%
AggC		58.38%

Table 5.29: Performance of gender identification on SmallEngDat using 0.5% as T_{min} threshold.

5.2.2.7 Authorship Identification

The same experiments that were performed in the previous section are now performed using the author name labels. This experiment is more challenging since the goal is to distinguish the writings of 11 different authors. In this experiment, the value of the T_{min} threshold is essential. If the number of estimated zones is lower than the number of classes, the performance of the algorithm is worse.

Table 5.30 shows the performance of our algorithm using different sets of features and 1% as T_{min}.

It can be observed that the features that worked better in the previous experiment lead now to lower accuracies due to an under-predicting of the number of clusters. As illustrated by Table 5.31, the performance improves significantly when the number of zones is higher (with the T_{min} threshold being set to 0.5%). As already for gender identification, the figures are contrasted with the performance of the baselines on this dataset.

5.2.2.8 Experiments on Public Datasets

In this experiment, we assess the performance of the algorithm using publicly available data already discussed in Section 5.2.2.4, for which the feature values and labels of each dataset instance are provided. These datasets stem from fields

Feats	#Zones	Accuracy
Full Set	3	37.45%
Character-Based	4	34.76%
Word-Based	7	40.23%
Sentence-Based	8	30.30%
Dictionary-Based	7	38.37%
Syntactic	3	39.29%
Char Word Dict	5	45.22%
Char Dict	5	40.84%
Char Syntax	4	46.65%
Char Dict Syntax	4	**49.46**%

Table 5.30: Performance of author identification on SmallEngDat using 1% as T_{min}.

Feats	#Zones	Accuracy
Full Set	8	65.98%
Character-Based	7	60.64%
Word-Based	13	44.41%
Sentence-Based	10	30.66%
Dictionary-Based	12	42.19%
Syntactic	12	46.12%
Char Word Dict	10	63.96%
Char Dict	10	55.83%
Char Syntax	9	62.62%
Char Dict Syntax	12	**69.38**%
ARK		61.01%
AKPP		63.85%
DBSCAN		22.31%
AggC		21.53%

Table 5.31: Performance of author identification on SmallEngDat using 0.5% as T_{min}.

that are very different from author profiling. Therefore, the performance of our algorithm on them can be taken as indicator of how well our algorithm performs in general.

The experiments on all datasets are performed using 1% as the T_{min} threshold.

Table 5.32 shows the performance of our algorithm, the predicted number of

zones, the number of classes that each of the datasets has as well as the accuracy of the baselines.

Dataset	#Zones	#Classes	Our Model	ARK	AKPP	DBSCAN	AggC
Spam	8	2	**72.68%**	66.21%	67.25%	67.94%	68.07%
Image	5	7	64.60%	65.91%	**68.68%**	26.36%	17.66%
Phishing	6	2	**82.02%**	66.55%	72.26%	66.53%	66.41%
Bank	5	2	**82.37%**	57.41%	55.47%	54.79%	54.52%
Cardio3	8	3	**87.74%**	77.79%	77.75%	79.88%	82.59%
Cardio10	10	10	89.85%	86.36%	**92.68%**	23.68%	22.01%
Blood	5	2	**78.20%**	76.13%	76.85%	76.87%	76.73%

Table 5.32: Performance on public data.

5.2.2.9 Discussion

In the first experiment, we saw that the selected features successfully differentiated the writings of men and women, achieving accuracies of more than 80% with selected combinations of features. Lowering T_{min} leads to smaller zones and an increase of the accuracy, but there are also cases in which adding extra clusters does not help to increase the accuracy of the algorithm. In general, it can be stated that the improvement in accuracy occurs when features that are very effective in gender differentiation are used with lower values of the T_{min} threshold. In these cases, we obtain smaller more fine-grained clusters, where the instances that are grouped together are those that are stylistically most similar.

As already mentioned above, the second experiment is considerably more challenging since the goal is to differentiate between the writings of 11 different authors (having less than 200 texts per author). The results show that when a number of clusters is estimated that is lower than the real number of classes, the accuracy decreases. When the value of the T_{min} threshold is adjusted, the performance improves drastically. Table 5.31 also shows that when the full set of features is used, eight dense zones are estimated and the system groups instances of the same author together in more than 65% of the cases. This figure increases to more than 69% of the cases when character, dictionary and syntactic features are used.

In both experiments, our algorithm significantly outperforms the four baselines. In view of the K-means baselines and the Agglomerative clustering, this is an indication that a fixed number of clusters, can work against the performance of the system in terms of accuracy, if the instances are not easily separable.

The last experiment deals with datasets that are completely different from author profiling. Also, these datasets have different topologic characteristics: 'Cardio3', 'Cardio10' and 'Image' are multi-class (3,10 and 7 respectively) problems with separable data (the majority of instances of a class are spatially close to the other instances of the same class and distant from the other classes). On the other hand, 'Bank', 'Phishing', 'Blood' and 'Spam' are binary classification problems of data that is much harder to separate.

In this experiment, our algorithm also performs rather well compared to the baselines. In five out of seven cases, it outperforms the standard randomly initialized K-means as well as K-means++. It also consistently outperforms DBSCAN and Agglomerative clustering. Both algorithms struggle in the case of multi-class separable data.

The fact that that our algorithm achieves competitive results on data unrelated to author profiling and on different topological configurations demonstrates that it is effective in general and that it can be applied to any clustering problem that uses numerical features.

5.2.2.10 Feature Analysis

The experiments demonstrated that different subsets of features achieved different values of accuracy and estimated dense zones. Let us analyze which individual features are most relevant. To do so, we use the full set of features and the gender identification dataset, for which we obtained in the corresponding experiment an estimation of three dense zones and achieved an accuracy of 66.75%.

To see which individual features were the most distinctive ones, we plot the mean values of each feature for each of the clusters in Figure 5.12. The letters 'M' and 'F' followed by a number are the number of instances per class (and thus per cluster). Each axis of the graph plots the mean values of one of the features. The labels of each individual feature are omitted for clarity and the absence of a feature implies that it has the value zero (the features are plotted in a logarithmic scale). The plot draws the profile of each one of the clusters. It can be observed that there are feature values with clear differences between clusters.

We then plot some of the most distinctive features (the ones that had clear differences in their mean values in Figure 5.12) per cluster and gender. In Figure 5.13, we can see the differences between some of the most distinctive character-, word- and dictionary-based features. The red dashed line represents female and the blue continuous line, the male writers.

The plot shows that some of the most distinctive features are the usage of semicolons, question and exclamation marks, percentages, parenthesis, hyphens, curse words and first person singular ('firstS') and plural ('firstP') pronouns. There are apparent differences in the style of men and women and also between clusters.

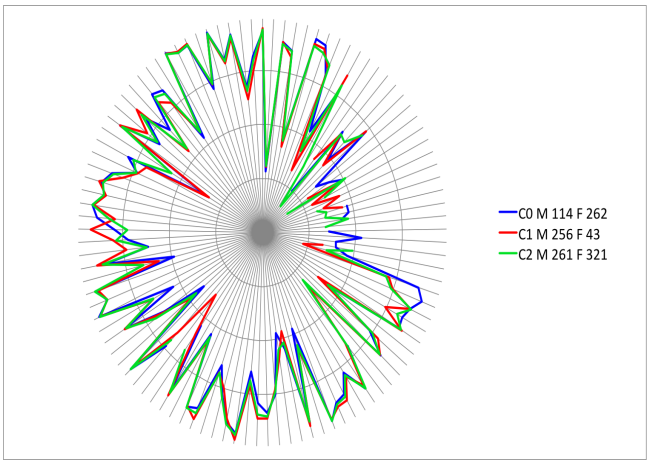


Figure 5.12: Feature mean values per cluster.

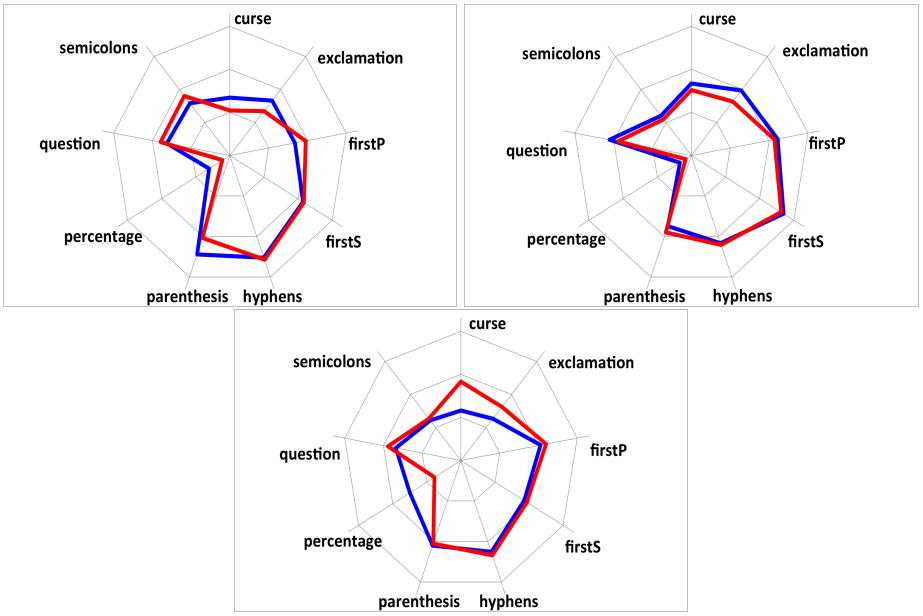


Figure 5.13: Values of character-, word-, and dictionary-based features per cluster and gender.

This suggests that each cluster captures a different kind of style for both genders.

The difference in the usage of the percentage sign can be related to the topic of the text (some of the texts are about finances and real estate), and the usage of curse words, exclamation marks and first person pronouns (both singular and plural) can be proportional to the implication of the writer in the narration they

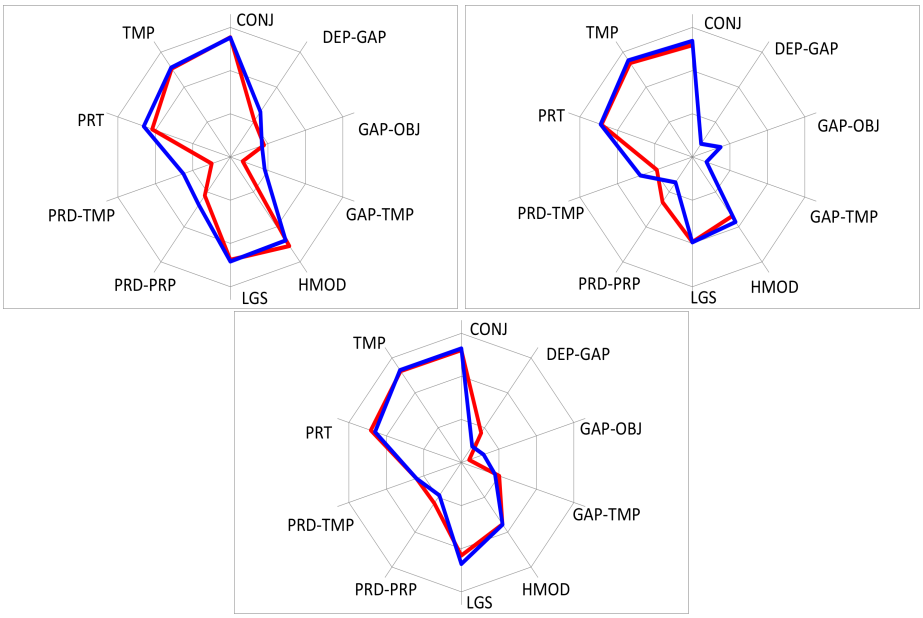


Figure 5.14: Syntactic mean feature values per cluster and gender.

authored. All these features (with the exception of the percentage mark) are likely to adopt different values, depending on how much the writer is either talking about him/herself, people that he/she is in touch with, or whether the writing is an opinion piece or a neutral narration.

Figure 5.14 displays the syntactic dependencies that are more distinctive between genders. The first cluster reveals significant differences between men and women in the usage of 'prd-tmp' and 'gap-obj' dependencies. Especially the latter gives us an indication of the complexity of the syntactic structures in the writing (more 'gap-obj' dependencies implies a higher complexity). In the second cluster, some of dependencies (such as 'conj', 'dep-gap', and 'gap-obj') are not reflected in the writings of the female authors. This might be due to the small number of female authors in this cluster (only 43) or due to the idiosyncrasy of the style. Further investigation is needed for a more assertive statement.

5.2.2.11 Instance Number Threshold Analysis

We saw how the modification of T_{min} influences the way the algorithm performs. In order to obtain further insight on this influence, we computed both the accuracy and the estimated number of zones for different values of T_{min} using the full set of features in the gender and authorship datasets. Figure 5.15 shows the effect of different values of T_{min} for the gender dataset.

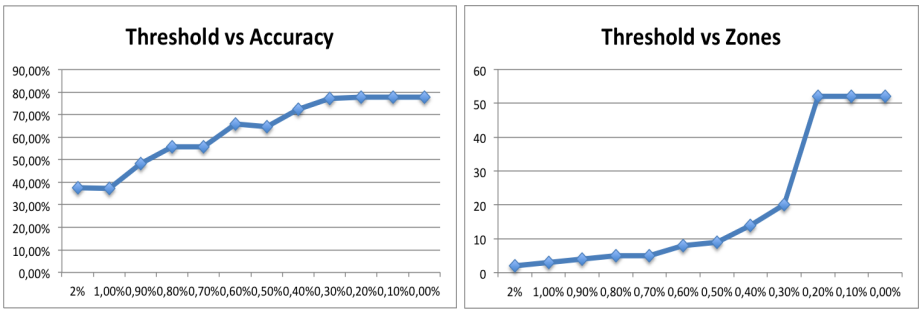


Figure 5.15: Effects of T_{min} in the gender dataset.

We can see a clear tendency in both graphs. The smaller the threshold, the higher the accuracy and the number of zones, and, as a result, smaller clusters. Before, we saw that in some feature combinations, an increase of the number of zones did not result in an increase of accuracy. It is interesting to note that the algorithm without restrictions (T_{min} = 0) in this respect outputs 52 dense zones for the 1257 instances that we have. The problem with this high number of zones is that the granularity of the clusters would be too high and their size too small. We would indeed have very similar instances together, but for some problems this could be too restrictive.

Figure 5.16 shows the same analysis on the authorship dataset.

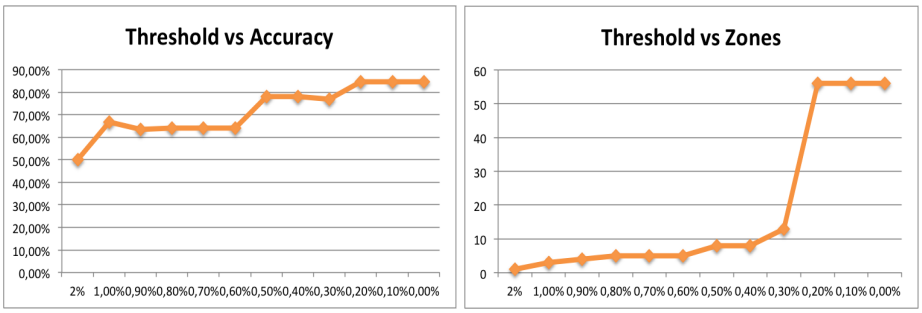


Figure 5.16: Effects of T_{min} in the authorship dataset.

The tendencies are very similar. The algorithm predicts a slightly higher number of dense zones when the threshold ranges between 0 and 0.20%, while from 0.20% onwards, the increase is much more drastic when compared to the gender graph. A similar tendency can be observed with the accuracy values, both tend to have higher accuracies with decreasing threshold.

5.2.2.12 Conclusions

We presented an unsupervised learning algorithm that automatically estimates the number of clusters to be formed during the clustering procedure. Experiments on gender and authorship identification datasets led to competitive results, compared to other common techniques which require higher parametrization. Experiments on publicly available data from other fields have been used to demonstrate that our algorithm is not restricted to the task of author profiling, but, rather, can be used as a general clustering algorithm. We analyzed the impact that different types of features have on the quality of author profiling, and the effect that the threshold for the number of instances in a cluster may have.

Chapter 6

CONCLUSIONS

In this final chapter we first present a short summary of what has been accomplished with this thesis. After that, we analyze the shortcomings of our approach and outline our future work. Finally, we list the scientific papers that have been published during the development of the thesis and the mentions in the media.

6.1 Summary

Several goals have been accomplished with this thesis. First of all, we were able to develop a small feature set composed mainly of syntactic and discourse-based features, which led to a very competitive performance of standard machine learning techniques. Even though in author profiling and identification, syntactic features are often mentioned, our combination of PoS, dependency and shape features is novel and helps characterize the text structure in a deep-linguistic manner. On the other hand, discourse features have never been used for the tasks of author profiling and author identification before.

The described feature set proved to be very versatile; it was applied to several tasks, namely gender, age, origin, book, author and language identification. In each task, our approach was able to achieve state-of-the-art performance. We also proved that the feature set is effective using different text genres as input data.

The experiments were mainly supervised learning applications using texts written in English, but semi-supervised, unsupervised and multilingual experiments were also implemented.

We have published all the compiled datasets in `https://github.com/joanSolCom/Datasets`. The code used to extract the features that are described in this thesis is also available in `https://github.com/joanSolCom/author_profiling_tools`.

6.2 Shortcomings of the Presented Thesis

This work has several shortcomings. Firstly, we always worked on noise-free environments; to adapt the feature set to chat texts or social media, pre-processing steps might be required. Secondly, right now our approach is very tool-dependent: a dependency parser and a discourse parser are required. Even though there are plenty of options to perform dependency parsing in different languages, discourse parsers are not available (to the best of our knowledge) for other languages. The dictionaries that are used to identify sensation and polar words would also need to be replaced with corresponding resources for other languages. As a result of this lack of resources for other languages, multilingual author profiling and identification is a challenge yet to be tackled. Thirdly, multiple genre scenarios have not been considered (i.e., training with a specific text genre, to then test with a different one), but theoretically, our focus on syntax and discourse can help in this sort of experiments. Finally, no systematic in-depth study on which machine learning techniques are best for the tasks of author profiling and identification has been carried out.

6.3 Future Work

Many possible extensions and alternative applications are planned to be in our future work.

We plan to adapt our methods to noisier environments, adding preprocessing steps to our usual pipeline to further generalize our methods and to apply them to chat texts and social media. Another future line of work that we plan to explore is literary stylistic variation, in which every novel of an author is analyzed to see the stylistic evolution of the author. This sort of study can be extremely insightful if applied to writers such as Terry Pratchett, who was diagnosed with Alzheimer disease during his literary career.

We also plan to expand the feature set. Deep syntax and communicative structure will also be considered as completely novel features in the field. Discourse features can be expanded, taking into account the inner structure of relations and the characteristics of nuclei and satellites. Semantic parsing is another promising source of features that is also in our plans. All of these structural features will contribute in future cross-genre experiments.

Systematic studies on which learning technique (considering machine learning and deep learning methods) is best for author profiling and identification will be carried out.

Some alternative applications where our methods can be applied are hate speech and automatic troll detection. Hate speech detection is about automatically detect-

ing hateful comments towards a certain group (e.g., racism, sexism, etc.). Automatic troll detection is the task of detecting harmful users in forums. We believe that our approach can be directly applied to these tasks effectively.

Other demographic traits will be considered in the future, such as sexual orientation. Since there are no publicly available resources that can be used for this purpose, we plan to construct and publish such a resource. After constructing this resource, sexual orientation detection can be performed. This task can be applied to improve marketing studies in order to further improve, e.g., the advertisement strategies of online companies.

Further multilingual approaches are also in our future plans. A possible approach that we will consider is to perform multilingual author profiling using Universal Dependencies (Nivre et al., 2016) to implement a language-independent approach. This set of dependencies generalizes language-specific dependencies with the goal to create a universal set, but is applicable in multilingual scenarios.

6.4 Publications and Media Mentions

After enumerating the achieved goals, let us introduce the list of publications that were published during the development of this thesis:

- Soler-Company, J. and Wanner, L. (2014). How to use less features and reach better performance in author gender identification. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, pages 26–31.

- Verhoeven, B., Soler-Company, J., and Daelemans, W. (2014). Evaluating Content-Independent Features for Personality Recognition. In *Proceedings of the 2014 ACM Multi Media on Workshop on Computational Personality Recognition*, pages 7–10.

- Soler-Company, J. and Wanner, L. (2015). Multiple Language Gender Identification for Blog Posts. In *Proceedings of the 37th Annual Cognitive Science Society Meeting (COGSCI)*, pages 2248–2253.

- Soler-Company, J., Ballesteros, M., Bohnet, B., Mille, S., and Wanner, L. (2015). Visualizing deep-syntactic parser output. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 56–60.

- Soler-Company, J. and Wanner, L. (2016a). A Semi-Supervised Approach for Gender Identification. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*, pages 146–153.

- Soler-Company, J. and Wanner, L. (2016b). Authorship Attribution Using Syntactic Dependencies. In *19th International Conference of the Catalan Association for Artificial Intelligence (CCIA)*, pages 303–308.

- Soler-Company, J. and Wanner, L. (2016c). Use of Discourse and Syntactic Features for Gender Identification. In *The Eighth Starting Artificial Intelligence Research Symposium. Collocated with the 22nd European Conference on Artificial Intelligence*, pages 215–220. Best poster award.

- Soler-Company, J. and Wanner, L. (2017a). Author and Gender Identification using Syntactic Dependencies and Discourse Relations. *Special Issue on Machine Learning and Applications in Artificial Intelligence. Pattern Recognition Letters*. (submitted).

- Soler-Company, J. and Wanner, L. (2017b). On the Relevance of Syntactic and Discourse Features for Author Profiling and Identification. In *15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 681–687.

Our work has also been mentioned in the media:

- (in Spanish) Los textos escritos delatan el sexo.

  `http://www.lavanguardia.com/cultura/20161017/411054860677/linguistica-estudio\-determina-sexo-autor-articulo-upf.html`.

- (in Catalan) El sexe determina el text.

  `https://menorca.info/menorca/vivir-menorca/2016/586906/sexe-determina-text.html`.

- (in Catalan) El sexe dels textos.

  `http://www.elpuntavui.cat/societat/article/5-societat/1012850-el-sexe-dels-textos.html`.

Bibliography

Abbasi, A. and Chen, H. (2005). Applying authorship analysis to extremist-group Web forum messages. *IEEE Intelligent Systems*, 20(5):67–75.

Adams, S. H. (1996). Statement analysis: What do suspects' words really reveal. *FBI Law Enforcement Bulletin*.

Aljumily, R. (2015). Hierarchical and Non-Hierarchical Linear and Non-Linear Clustering Methods to "Shakespeare Authorship Question". *Social Sciences*, 4(3):758–799.

Allison, B. and Guthrie, L. (2008). Authorship Attribution of E-Mail: Comparing Classifiers over a New Corpus for Evaluation. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC-2008)*.

Altheneyan, A. S. and Menai, M. E. B. (2014). Naïve bayes classifiers for authorship attribution of arabic texts. *Journal of King Saud University-Computer and Information Sciences*, 26(4):473–484.

Amuchi, F., Al-Nemrat, A., Alazab, M., and Layton, R. (2012). Identifying cyber predators through forensic authorship analysis of chat logs. In *Cybercrime and Trustworthy Computing Workshop (CTC), 2012 Third*, pages 28–37. IEEE.

Argamon, S., Dhawle, S., Koppel, M., and Pennebaker, J. W. (2005). Lexical predictors of personality type. In *Proceedings of the 2005 Joint Annual Meeting of the Interface and the Classification Society of North America*.

Argamon, S. and Juola, P. (2011). Overview of the International Authorship Identification Competition at PAN-2011. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*.

Argamon, S., Koppel, M., Pennebaker, J. W., and Schler, J. (2007). Mining the Blogosphere: Age, gender and the varieties of self-expression. *First Monday*, 12(9).

Argamon, S., Koppel, M., Pennebaker, J. W., and Schler, J. (2009). Automatically Profiling the Author of an Anonymous Text. *Commun. ACM*, 52(2):119–123.

Arthur, D. and Vassilvitskii, S. (2007). K-means++: The Advantages of Careful Seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Arun, Saradha, Suresh, V., Murty, and Veni Madhavan, C. E. (2009). Stopwords and Stylometry : A Latent Dirichlet Allocation Approach. In *NIPS Workshop on Applications for Topic Models: Text and Beyond*.

Barbieri, F. and Saggion, H. (2014a). Automatic Detection of Irony and Humour in Twitter. In *ICCC-2014, the 5th International Conference on Computational Creativity, Ljubljana, Slovenia, June 2014*.

Barbieri, F. and Saggion, H. (2014b). Modelling Irony in Twitter. In *EACL*, pages 56–64.

Barbieri, F. and Saggion, H. (2014c). Modelling Irony in Twitter: Feature Analysis and Evaluation. In *Language Resources and Evaluation Conference, LREC*.

Bayot, R. K. and Goncalves, T. (2016). Author profiling using svms and word embedding averages. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*.

Bennett, W. R. (1976). *Scientific and engineering problem-solving with the computer*. Prentice Hall PTR.

Biber, D. and Finegan, E. (1989). Drift and the evolution of English style: A history of three genres. *Language*, pages 487–517.

Bohnet, B. and Nivre, J. (2012). A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1455–1465. Association for Computational Linguistics.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Burger, J. D., Henderson, J., Kim, G., and Zarrella, G. (2011). Discriminating Gender on Twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1301–1309, Stroudsburg, PA, USA. Association for Computational Linguistics.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167.

Burstein, J., Marcu, D., and Knight, K. (2003). Finding the WRITE Stuff: Automatic Identification of Discourse Structure in Student Essays. *IEEE Intelligent Systems*, 18(1):32–39.

Burton, K., Java, A., and Soboroff, I. (2009). The icwsm 2009 spinn3r dataset. In *Proceedings of the Third Annual Conference on Weblogs and Social Media (ICWSM 2009), San Jose, CA*.

Calix, K., Connors, M., Levy, D., Manzar, H., and Westcott, S. (2008). Stylometry for e-mail author identification and authentication. In *Proceedings of CSIS Research Day, Pace Universit*, pages 1048–1054.

Cambria, E., Havasi, C., and Hussain, A. (2012). SenticNet 2: A Semantic and Affective Resource for Opinion Mining and Sentiment Analysis. In *Proceedings of the Florida Artificial Intelligence Research Society conference* , pages 202–207.

Can, F. and Patton, J. M. (2004). Change of writing style with time. *Computers and the Humanities*, 38(1):61–82.

Castillo, E., Cervantes, O., Pinto, D., and León, S. (2014). Unsupervised Method for the Authorship Identification Task. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, pages 877–897, Sheffield, UK. CEUR-WS.org.

Charniak, E. and Johnson, M. (2005). Coarse-to-fine N-best Parsing and MaxEnt Discriminative Reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chen, X., Hao, P., Chandramouli, R., and Subbalakshmi, K. (2011). Authorship similarity detection from email messages. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 375–386. Springer.

Cheng, N., Chandramouli, R., and Subbalakshmi, K. P. (2011). Author Gender Identification from Text. *Digit. Investig.*, 8(1):78–88.

Cheng, N., Chen, X., Chandramouli, R., and Subbalakshmi, K. P. (2009). Gender identification from E-mails. In *Computational Intelligence and Data Mining, 2009. CIDM '09. IEEE Symposium on*, pages 154–158.

Chklovski, T. and Mihalcea, R. (2002). Building a sense tagged corpus with open mind word expert. In *Proceedings of the ACL-02 workshop on Word sense disambiguation: recent successes and future directions-Volume 8*, pages 116–122. Association for Computational Linguistics.

Clear, J. H. (1993). *The Digital Word*. MIT Press, Cambridge, MA, USA.

Crystal, D. and Davy, D. (1969). *Investigating English Style*. Longman Group Ltd., London.

David, C. C., Miller, D., and Walker, K. (2004). The Fisher Corpus: a Resource for the Next Generations of Speech-to-Text. In *Proceedings 4th International Conference on Language Resources and Evaluation*, pages 69–71.

De Vel, O., Anderson, A., Corney, M., and Mohay, G. (2001). Mining e-Mail Content for Author Identification Forensics. *SIGMOD Rec.*, 30(4):55–64.

DiMarco, C. and Hirst, G. (1993). A Computational Theory of Goal-Directed Style in Syntax . *Computational Linguistics*, 19(3):451–499.

Eiselt, M. P. B. S. A. and Rosso, A. B.-C. P. (2009). Overview of the 1st international competition on plagiarism detection. In *3rd PAN Workshop. Uncovering Plagiarism, Authorship and Social Software Misuse*, page 1.

Estival, D., Gaustad, T., Hutchinson, B., Pham, S. B., and Radford, W. (2007). Author profiling for English emails. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pages 263–272.

Feng, S., Banerjee, R., and Choi, Y. (2012). Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 171–175. Association for Computational Linguistics.

Ferilli, S., Redavid, D., and Esposito, F. (2015). Unsupervised Author Identification and Characterization. In *Italian Research Conference on Digital Libraries*, pages 129–141. Springer.

Fissette, M. (2010). Author Identifcation in Short Texts. Bachelor's Thesis.

Frantzeskou, G., Stamatatos, E., Gritzalis, S., and Chaski, C. E. (2007). Identifying authorship by byte-level n-grams: The source code author profile (scap) method. *International Journal of Digital Evidence*, pages 1–18.

Freeman, D. C. (1970). *Linguistics and literary style*. Holt Rinehart & Winston.

Gamon, M. (2004). Linguistic correlates of style: authorship classification with deep linguistic analysis features. In *Proceedings of the 20th international conference on Computational Linguistics*, page 611. Association for Computational Linguistics.

Ghaeini, M. (2013). Intrinsic author identification using modified weighted knn. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*. CEUR-WS.org.

Goldberg, L. R. (1993). The structure of phenotypic personality traits. *American Psychologist*, 48(1):26–34.

Goswami, S., Sarkar, S., and Rustagi, M. (2009). Stylometric Analysis of Bloggers' Age and Gender. In *ICWSM*. The AAAI Press.

Granger, S. (2003). The International Corpus of Learner English: A New Resource for Foreign Language Learning and Teaching and Second Language Acquisition Research. *TESOL Quarterly*, 37(3):538–546.

Groom, C. J. and Pennebaker, J. W. (2005). The Language of Love: Sex, Sexual Orientation, and Language Use in Online Personal Advertisements. *Sex Roles*, 52(7):447–461.

Gupta, A., Kumaraguru, P., and Sureka, A. (2012). Characterizing pedophile conversations on the internet using online grooming. *arXiv preprint arXiv:1208.4324*.

Haji Mohammad, S., Kelly P, D., Susan, B., and Derek, R. (2016). A Web of Hate: Tackling Hateful Speech in Online Social Spaces. In *Proceedings of the Workshop on Text Analytics for Cybersecurity and Online Safety (TA-COS 2016)*, Portorož, Slovenia. ELRA, ELRA.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Halliday, M. A. K. and Matthiessen, C. M. I. M. (2004). *An introduction to functional grammar / M.A.K. Halliday* . Hodder Arnold London, 3rd ed. / rev. by christian m.i.m. matthiessen. edition.

Harman, D. (1994). Overview of the second text retrieval conference (TREC-2). In *Proceedings of the Workshop on Human Language Technology*, pages 351–357. Association for Computational Linguistics.

Havasi, C., Speer, R., and Alonso, J. (2007). ConceptNet 3: a flexible, multilingual semantic network for common sense knowledge. In *Recent advances in natural language processing (RANLP)*, pages 27–29.

Henderson, J., Zarrella, G., Pfeifer, C., and Burger, J. D. (2013). Discriminating Non-Native English with 350 Words. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*. The Association for Computational Linguistics.

Heylighen, F. and Dewaele, J.-M. (2002). Variation in the Contextuality of Language: An Empirical Measure. *Foundations of Science*, 7(3):293–340.

Holmes, D. I. (1985). The analysis of literary style–a review. *Journal of the Royal Statistical Society. Series A (General)*, pages 328–341.

Holmes, J. and Meyerhoff, M. (2008). *The handbook of language and gender*, volume 25. John Wiley & Sons.

Houvardas, J. and Stamatatos, E. (2006). N-gram feature selection for authorship identification. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pages 77–86. Springer.

Hu, M. and Liu, B. (2004). Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Ikeda, D., Takamura, H., and Okumura, M. (2008). Semi-Supervised Learning for Blog Classification. In *22nd AAAI Conference on Artificial Intelligence*. AAAI.

Ikeda, K., Hattori, G., Ono, C., Asoh, H., and Higashino, T. (2013). Twitter user profiling based on text and community mining for market analysis. *Knowledge-Based Systems*, 51:35–47.

Inches, G. and Crestani, F. (2012). Overview of the International Sexual Predator Identification Competition at PAN-2012. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means . *Pattern Recognition Letters*, 31(8):651 – 666. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR)19th International Conference in Pattern Recognition (ICPR).

Juola, P. (2012). An Overview of the Traditional Authorship Attribution Subtask. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*.

Juola, P. and Stamatatos, E. (2013). Overview of the Author Identification Task at PAN 2013. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*.

Kahane, S. (2003). The meaning-text theory. *Dependency and Valency. An International Handbook of Contemporary Research*, 1:546–570.

Kapovciute-Dzikiene, J., Utka, A., and Sarkute, L. (2015). Authorship attribution and author profiling of Lithuanian literary texts. In *Proceedings of the 5th workshop on Balto-Slavic Natural Language Processing: Hissar, Bulgaria, 10–11 September 2015, p. 96-105*.

Keogh, E. and Mueen, A. (2011). Curse of dimensionality. In *Encyclopedia of Machine Learning*, pages 257–258. Springer.

Kevselj, V., Peng, F., Cercone, N., and Thomas, C. (2003). N-gram-based author profiles for authorship attribution. In *Proceedings of the Conference Pacific Association for Computational Linguistics, PACLING'03*, pages 255–264.

Khan, A. M. R. (2012). A simple but powerful e-mail authorship attribution system. In *International Conference on Machine Learning and Computing*.

Klein, D. and Manning, C. D. (2003). Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Stroudsburg, PA, USA. Association for Computational Linguistics.

Koppel, M., Argamon, S., and Shimoni, A. (2003). Automatically Categorizing Written Texts by Author Gender.

Koppel, M., Schler, J., and Zigdon, K. (2005). Determining an Author's Native Language by Mining a Text for Errors. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 624–628, New York, NY, USA. ACM.

Kotsiantis, S. (2007). Supervised Machine Learning: A Review of Classification Techniques. *Informatica*, 31:249–268.

Kourtis, I. and Stamatatos, E. (2011). Author identification using semi-supervised learning. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*.

Kucukyilmaz, T., Cambazoglu, B. B., Aykanat, C., and Can, F. (2006). Chat Mining for Gender Prediction. In *Proceedings of the 4th International Conference on Advances in Information Systems*, ADVIS'06, pages 274–283, Berlin, Heidelberg. Springer-Verlag.

Lakoff, G. (1973). Hedges: A study in meaning criteria and the logic of fuzzy concepts. *Journal of philosophical logic*, 2(4):458–508.

Lavelle, E. (1997). Writing style and the narrative essay. *British Journal of Educational Psychology*, 67(4):475–482.

Leech, G. (2007). Style in fiction revisited: the beginning of Great Expectations. *Style*, 41(2):117–132.

Leech, G. N. and Short, M. (2007). *Style in fiction: A linguistic introduction to English fictional prose*. Pearson Education.

Leuzzi, F., Ferilli, S., and Rotella, F. (2013). A relational unsupervised approach to author identification. In *International Workshop on New Frontiers in Mining Complex Patterns*, pages 214–228. Springer.

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). RCV1: A New Benchmark Collection for Text Categorization Research. *J. Mach. Learn. Res.*, 5:361–397.

Lichman, M. (2013). UCI Machine Learning Repository.

Luyckx, K. and Daelemans, W. (2008a). Authorship attribution and verification with many authors and limited data. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 513–520. Association for Computational Linguistics.

Luyckx, K. and Daelemans, W. (2008b). Personae: A corpus for author and personality prediction from text. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC-2008)*.

Luyckx, K. and Daelemans, W. (2008c). Using Syntactic Features to Predict Author Personality from Text. *Presented at Digital Humanities 2008, Oulu, Finland*.

Madsen, R. E., Kauchak, D., and Elkan, C. (2005). Modeling word burstiness using the Dirichlet distribution. In *Proceedings of the 22nd international conference on Machine learning*, pages 545–552. ACM.

Maharjan, S., Shrestha, P., and Solorio, T. (2014). A Simple Approach to Author Profiling in MapReduce. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, Sheffield, UK. CEUR-WS.org.

Maitra, P., Ghosh, S., and Das, D. (2014). Authorship Verification–An Approach based on Random Forest. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*.

Makazhanov, A. and Rafiei, D. (2013). Predicting Political Preference of Twitter Users. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '13, pages 298–305, New York, NY, USA. ACM.

Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Matthews, R. A. and Merriam, T. V. (1993). Neural computation in stylometry I: An application to the works of Shakespeare and Fletcher. *Literary and Linguistic Computing*, 8(4):203–209.

Meina, M., Brodzinska, K., Celmer, B., Czoków, M., Patera, M., Pezacki, J., and Wilk, M. (2013). Ensemble-based classification for author profiling using various features. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*.

Melcuk, I. A. (1988). *Dependency syntax: theory and practice*. SUNY press.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Modaresi, P. and Gross, P. (2014). A Language Independent Author Verifier Using Fuzzy C-Means Clustering. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, pages 877–897, Sheffield, UK. CEUR-WS.org.

Mukherjee, A. and Liu, B. (2010). Improving Gender Classification of Blog Authors. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 207–217, Stroudsburg, PA, USA. Association for Computational Linguistics.

Myers, I. and Myers, P. (2010). *Gifts differing: Understanding personality type*. Nicholas Brealey Publishing.

Nerbonne, J. (2013). The Secret Life of Pronouns. What Our Words Say About Us. *Literary and Linguistic Computing*, page fqt006.

Newman, M. L., Pennebaker, J. W., Berry, D. S., and Richards, J. M. (2003). Lying words: Predicting deception from linguistic styles. *Personality and social psychology bulletin*, 29(5):665–675.

Nguyen, D., Gravel, R., Trieschnigg, D., and Meder, T. (2013). "How Old Do You Think I Am?" A Study of Language and Age in Twitter. In *ICWSM*. The AAAI Press.

Nguyen, D., Smith, N. A., and Rosé, C. P. (2011). Author Age Prediction from Text Using Linear Regression. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, LaTeCH '11, pages 115–123, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., et al. (2016). Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1659–1666.

Oberlander, J. and Nowson, S. (2006). Whose thumb is it anyway?: classifying author personality from weblog text. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 627–634. Association for Computational Linguistics.

Otterbacher, J. (2010). Inferring Gender of Movie Reviewers: Exploiting Writing Style, Content and Metadata. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 369–378, New York, NY, USA. ACM.

Palomino-Garibay, A., Camacho-González, A. T., Fierro-Villaneda, R. A., Hernández-Farias, I., Buscaldi, D., and Meza-Ruiz, I. V. (2015). A random forest approach for authorship profiling. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*.

Pearl, L. and Steyvers, M. (2012). Detecting authorship deception: a supervised machine learning approach using author writeprints. *Literary and linguistic computing*, page fqs003.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peersman, C., Daelemans, W., and Van Vaerenbergh, L. (2011). Predicting Age and Gender in Online Social Networks. In *Proceedings of the 3rd International Workshop on Search and Mining User-generated Contents*, SMUC '11, pages 37–44, New York, NY, USA. ACM.

Pennebaker, J. W., Francis, M. E., and Booth, R. J. (2001). *Linguistic Inquiry and Word Count*. Lawerence Erlbaum Associates, Mahwah, NJ.

Pennebaker, J. W., Mehl, M. R., and Niederhoffer, K. G. (2003). Psychological aspects of natural language use: Our words, our selves. *Annual review of psychology*, 54(1):547–577.

Pham, D. D., Tran, G. B., and Pham, S. B. (2009). Author Profiling for Vietnamese Blogs. In *Asian Language Processing, 2009. IALP '09. International Conference on*, pages 190–194.

Poria, S., Gelbukh, A., Agarwal, B., Cambria, E., and Howard, N. (2013a). *Common Sense Knowledge Based Personality Recognition from Text*, pages 484–496. Springer Berlin Heidelberg, Berlin, Heidelberg.

Poria, S., Gelbukh, A., Hussain, A., Howard, N., Das, D., and Bandyopadhyay, S. (2013b). Enhanced SenticNet with affective labels for concept-based opinion mining. *IEEE Intelligent Systems*, 28(2):31–38.

Potthast, M., Eiselt, A., Barrón Cedeño, L. A., Stein, B., and Rosso, P. (2011). Overview of the 3rd international competition on plagiarism detection. In *CEUR Workshop Proceedings*. CEUR Workshop Proceedings.

Potthast, M., Hagen, M., Gollub, T., Tippmann, M., Kiesel, J., Rosso, P., Stamatatos, E., and Stein, B. (2013). Overview of the 5th international competition on plagiarism detection. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, pages 301–331. CELCT.

Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

Raghavan, S., Kovashka, A., and Mooney, R. (2010). Authorship Attribution Using Probabilistic Context-free Grammars. In *Proceedings of the ACL 2010*

Conference Short Papers, ACLShort '10, pages 38–42, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rangel, F., Hernández, I., Rosso, P., and Reyes, A. (2014a). Emotions and Irony per Gender in Facebook. In *ES3LOD 2014 5th International Workshop on Emotion, Social Signals, Sentiment and Linked Open Data Proceedings*, pages 68–73.

Rangel, F. and Rosso, P. (2013a). On the identification of emotions and authors' gender in facebook comments on the basis of their writing style. In *CEUR Workshop Proceedings*, volume 1096, pages 34–46. CEUR Workshop Proceedings.

Rangel, F. and Rosso, P. (2013b). Use of Language and Author Profiling: Identification of Gender and Age. *Natural Language Processing and Cognitive Science*, page 177.

Rangel, F., Rosso, P., Moshe Koppel, M., Stamatatos, E., and Inches, G. (2013). Overview of the author profiling task at pan 2013. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, pages 352–365. CELCT.

Rangel, F., Rosso, P., Potthast, M., Stein, B., and Daelemans, W. (2015). Overview of the 3rd Author Profiling Task at PAN 2015. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*.

Rangel, F., Rosso, P., Potthast, M., Trenkmann, M., Stein, B., Verhoeven, B., Daeleman, W., et al. (2014b). Overview of the 2nd author profiling task at pan 2014. In *CEUR Workshop Proceedings*, volume 1180, pages 898–927. CEUR Workshop Proceedings.

Rangel, F., Rosso, P., Verhoeven, B., Daelemans, W., Potthast, M., and Stein, B. (2016). Overview of the 4th author profiling task at PAN 2016: cross-genre evaluations. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*.

Rish, I. (2001). An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM New York.

Robinson, J. J. (1970). Dependency structures and transformational rules. *Language*, pages 259–285.

Rosenthal, S. and McKeown, K. (2011). Age Prediction in Blogs: A Study of Style, Content, and Online Behavior in Pre- and Post-social Media Generations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 763–772, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sapkota, U., Solorio, T., Montes-y Gómez, M., and Rosso, P. (2013). *The Use of Orthogonal Similarity Relations in the Prediction of Authorship*, pages 463–475. Springer Berlin Heidelberg, Berlin, Heidelberg.

Sarawgi, R., Gajulapalli, K., and Choi, Y. (2011). Gender Attribution: Tracing Stylometric Evidence Beyond Topic and Genre. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, CoNLL '11, pages 78–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

Schler, J., Koppel, M., Argamon, S., and Pennebaker, J. W. (2006). Effects of Age and Gender on Blogging. In *Proceedings of the AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 199–205. AAAI.

Schneider, C. A. (2005). *Willa Cather's "O Pioneers!" as a response to Kate Chopin's "The Awakening"*. PhD thesis, University of Nebraska - Lincoln.

Schwartz, R., Tsur, O., Rappoport, A., and Koppel, M. (2013). Authorship attribution of micro-messages. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1880–1891, Seattle, Washington, USA. Association for Computational Linguistics.

Shetty, J. and Adibi, J. (2004). The Enron email dataset database schema and brief statistical report. *Information sciences institute technical report, University of Southern California*, 4.

Sidorov, G., Miranda-Jiménez, S., Viveros-Jiménez, F., Gelbukh, A., Castro-Sánchez, N., Velásquez, F., Díaz-Rangel, I., Suárez-Guerra, S., Treviño, A., and Gordon, J. (2013). Empirical Study of Machine Learning Based Approach for Opinion Mining in Tweets. In *Proceedings of the 11th Mexican International Conference on Advances in Artificial Intelligence - Volume Part I*, MICAI'12, pages 1–14, Berlin, Heidelberg. Springer-Verlag.

Slatcher, R. B., Chung, C. K., Pennebaker, J. W., and Stone, L. D. (2007). Winning words: Individual differences in linguistic style among US presidential and vice presidential candidates. *Journal of Research in Personality*, 41(1):63–75.

Soler-Company, J., Ballesteros, M., Bohnet, B., Mille, S., and Wanner, L. (2015). Visualizing deep-syntactic parser output. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 56–60.

Soler-Company, J. and Wanner, L. (2014). How to use less features and reach better performance in author gender identification. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, pages 26–31.

Soler-Company, J. and Wanner, L. (2015). Multiple Language Gender Identification for Blog Posts. In *Proceedings of the 37th Annual Cognitive Science Society Meeting (COGSCI)*, pages 2248–2253.

Soler-Company, J. and Wanner, L. (2016a). A Semi-Supervised Approach for Gender Identification. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*, pages 146–153.

Soler-Company, J. and Wanner, L. (2016b). Authorship Attribution Using Syntactic Dependencies. In *19th International Conference of the Catalan Association for Artificial Intelligence (CCIA)*, pages 303–308.

Soler-Company, J. and Wanner, L. (2016c). Use of Discourse and Syntactic Features for Gender Identification. In *The Eighth Starting Artificial Intelligence Research Symposium. Collocated with the 22nd European Conference on Artificial Intelligence*, pages 215–220.

Soler-Company, J. and Wanner, L. (2017a). Author and Gender Identification using Syntactic Dependencies and Discourse Relations. *Special Issue on Machine Learning and Applications in Artificial Intelligence. Pattern Recognition Letters*.

Soler-Company, J. and Wanner, L. (2017b). On the Relevance of Syntactic and Discourse Features for Author Profiling and Identification. In *15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 681–687.

Staiano, J. and Guerini, M. (2014). DepecheMood: a Lexicon for emotion analysis from crowd-annotated news. *CoRR*, abs/1405.1605.

Stamatatos, E., Daelemans, W., Verhoeven, B., Stein, B., Potthast, M., Juola, P., Sanchez-Perez, M. A., and Barrón-Cedeño, A. (2014). Overview of the Author Identification Task at PAN 2014. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, pages 877–897.

Stamatatos, E., Fakotakis, N., and Kokkinakis, G. (2001). Computer-based Authorship Attribution without Lexical Measures. In *Computers and the Humanities*, pages 193–214.

Stamatatos, E., Potthast, M., Rangel, F., Rosso, P., and Stein, B. (2015). Overview of the PAN/CLEF 2015 evaluation lab. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, pages 518–538. Springer.

Surdeanu, M., Hicks, T., and Valenzuela-Escárcega, M. A. (2015). Two Practical Rhetorical Structure Theory Parsers. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT): Software Demonstrations*.

Surdeanu, M., Johansson, R., Meyers, A., Màrquez, L., and Nivre, J. (2008). The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177. Association for Computational Linguistics.

Tetreault, J. R. and Blanchard, D. (2012). Native Tongues, Lost and Found: Resources and Empirical Evaluations in Native Language Identification. In *24th International Conference on Computational Linguistics*.

Tofighi, P., Köse, C., and Rouka, L. (2012). Author's Native Language Identification from Web-Based Texts. *International Journal of Computer and Communication Engineering*, 1(1):47.

Tulkens, S., Hilte, L., Lodewyckx, E., Verhoeven, B., and Daelemans, W. (2016). A Dictionary-based Approach to Racism Detection in Dutch Social Media. In *Proceedings of the Workshop on Text Analytics for Cybersecurity and Online Safety (TA-COS 2016)*, Portorož, Slovenia. ELRA, ELRA.

Van Halteren, H. (2004). Linguistic Profiling for Author Recognition and Verification. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Van Hee, C., Lefever, E., Verhoeven, B., Mennes, J., Desmet, B., De Pauw, G., Daelemans, W., and Hoste, V. (2015a). Automatic detection and prevention of cyberbullying. In *International Conference on Human and Social Analytics (HUSO 2015)*, pages 13–18. IARIA.

Van Hee, C., Lefever, E., Verhoeven, B., Mennes, J., Desmet, B., De Pauw, G., Daelemans, W., and Hoste, V. (2015b). Detection and fine-grained classification of cyberbullying events. In *International Conference Recent Advances in Natural Language Processing (RANLP)*, pages 672–680.

Verhoeven, B., Daelemans, W., and De Smedt, T. (2013). Ensemble methods for personality recognition. In *Proceedings of WCPR13, in conjunction with ICWSM-13*.

Verhoeven, B., Soler-Company, J., and Daelemans, W. (2014). Evaluating Content-Independent Features for Personality Recognition. In *Proceedings of the 2014 ACM Multi Media on Workshop on Computational Personality Recognition*, pages 7–10.

Vrij, A. (2008). *Detecting lies and deceit: Pitfalls and opportunities*. John Wiley & Sons.

Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

Wilson, M. (1988). MRC Psycholinguistic Database: Machine-usable dictionary, version 2.00. *Behavior Research Methods, Instruments and Computers*, 20(1):6–10.

Wong, S.-M. J. and Dras, M. (2009). Contrastive analysis and native language identification. In *Proceedings of the Australasian Language Technology Association Workshop*, pages 53–61. Citeseer.

Wong, S.-M. J. and Dras, M. (2011). Exploiting Parse Structures for Native Language Identification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1600–1610, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yeh, I.-C., Yang, K.-J., and Ting, T.-M. (2009). Knowledge Discovery on RFM Model Using Bernoulli Sequence. *Expert Syst. Appl.*, 36(3):5866–5871.

Zamani, H., Esfahani, H. N., Babaie, P., Abnar, S., Dehghani, M., and Shakery, A. (2014). Authorship identification using dynamic selection of features from probabilistic feature set. In *PAN, collocated with CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, pages 877–897, Sheffield, UK. CEUR-WS.org.

Zhang, H. (2004). The optimality of naive Bayes. *AA*, 1(2):3.

Zhao, Y. and Zobel, J. (2005). *Effective and Scalable Authorship Attribution Using Function Words*, pages 174–189. Springer Berlin Heidelberg, Berlin, Heidelberg.

Zulfadhilah, M., Prayudi, Y., and Riadi, I. (2016). Cyber Profiling Using Log Analysis And K-Means Clustering. *International Journal of Advanced Computer Science & Applications*, 1(7):430–435.