Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Information Security and Communication Technology

Eirik Holbæk

Using Author Profiling to Determine the Age Group of an Author

Master's thesis in Communication Technology
Supervisor: Patrick Bours
June 2019

NTNU
Norwegian University of
Science and Technology

Title: Using Author Profiling to Determine the Age Group of an Author

Student: Eirik Holbæk

Problem description:

Author profiling can be defined as the task of determining one or more attribute of an author based on how they write. Among these attributes, the most common is to try to determine the traits like gender, age, place of origin and personal traits. The field of author profiling has seen a growth of interest in recent years. As it can be applied in many different use cases, for instance, marketing, increase internet security an in forensic investigation.

This thesis will centre around determining the age group of an author by analysing the text that is written. The main objective will be to determine if the author in a chatroom environment is a child (below the age of 18) or an adult (the age of 25 and above). The thesis will take in use the current state of the art author profiling methods as well as train a machine learning algorithm over a corpus and use the model to determine the age group (adult/child) of an author.

Responsible professor: Patrick Bours, IIK

Abstract

This thesis investigates how to determine the age group of an author, mainly if the author is a child, below the age of 18, or an adult, above the age of 25. Furthermore, the goal is to explore which textual features across different genres best correlate with the age of an author. Lastly, we want to investigate if a single model would be sufficient to predict age across various genres, or if the different domains need an individual model. To answer these questions, several data sets, previously used in author profiling research, have been collected. The data sets gathered contain blog texts, social media data and Twitter data. Furthermore, numerous experiments are implemented using commonly used machine learning classification algorithms and language recognition methods. The experiments are performed on individual genre data sets, as well as combined domains.

The results showed that it is possible to determine the age group of authors with relative accuracy, based on how they write. Results also reveal that the linear kernel SVM (Support Vector Machine) produces the best results throughout the experiments, in regards to overall prediction accuracy, precision and recall score, and the combined F_1 measure. Moreover, some of the textual features that are effective in distinguishing text written by the different age groups across the genres are TF-IDF (Term Frequency - Inverse Document Frequency), LIWC (Linguistic Inquiry and Word Count), n-grams, PoS (Part of Speech) tagging and stylistic language frequencies. Additionally, the results show that the models that are trained on a combined set of genres underperformed compared to models that trained only on a single domain.

Sammendrag

Denne masteroppgaven utforsker hvordan fastslå aldersgruppen til en forfatter. I hovedsak om forfatteren er et barn, som vil si under 18 år, eller voksen, 25 år og oppover. Videre er målet å undersøke hvilke tekstlige trekk som best korrelerer med alderen til en forfatter, over flere genrer. Til slutt, vil vi utforske om det vil være tilstrekkelig å kun bruke en felles modell for å predikere alderen over flere domener, eller om hver enkelt genre trenger en individuell modell. For å få svar på disse spørsmålene, har datasett fra tidligere forsking innenfor feltet forfatterprofilering, blitt samlet inn. Disse datasettene inneholder bloggdata, sosial mediatekster og Twitterdata. Videre har flere eksperimenter blitt utført på disse datasettene, der vi brukte maskinlærings algoritmer ofte brukt til klassifisering, samt ofte brukte språkgjenkjennelsesmetoder. Eksperimentene som ble utført ble gjort på individuelle datasett, i tillegg til kombinerte datasett.

Resultatene viser at det er mulig å fastslå aldersgruppen til forfattere basert på hvordan de skriver, med relativ høy treffsikkerhet. Videre viser også resultatene fra eksperimentene at lineær kernel SVM (Support Vector Machine) produserte de beste resultatene, med tanke på treffsikkerhet, presisjon og recall score, og den kombinerte F_1 verdien. Det erflere tekstlige trekk som ernyttige til å skille tekstene fra de forskjellige aldersgruppene og genere fra hverandre. Noen av disse er TF-IDF (Term Frequency - Inverse Document Frequency), LIWC (Linguistic Inquiry and Word Count), n-grams, PoS (Part of Speech) tagging og frekvensen stilistiske språklige trekk. Til slutt, viser resultatene at modellene som er trent på kombinerte sett med genre, gjorde det betraktelig dårligere enn modeller som bare var trent på individuelle domener.

Preface

This master thesis is submitted at the Department of Information Security and Communication Technology at Norwegian University of Science and Technology (NTNU). The thesis constitutes the final project for the MSc program in Communication Technology with specialisation in Information Security. The duration of the study has been 20 weeks, performed in the Spring of 2019.

I would like to thank my responsible professor Patrick Bours, for the weekly meetings, and with valuable advice and constructive feedback for the master thesis.

Trondheim, 6th of June 2019, Eirik Holbæk.

Contents

List of Figures

List of Tables

List of Acronyms

LDA Latent Dirichlet Allocation.

LIWC Linguistic Inquiry and WordCount.

NB Naive Bayes.

NLP Natural Language Processing.

NLTK Natural Language Toolkit.

PAN Plagiarism, Authorship and Social Software Misuse.

PoS Part of Speech.

RBF Radial Basis Function.

SVM Support Vector Machine.

TF-IDF Term frequency-inverse document frequency.

Chapter 1
Introduction

Author profiling can be defined as the task of determining one or more attribute of an author based on how they write. These attributes can be gender, age, personality traits. Author profiling should not be confused with author identification, where the goal is to identify the author from a closed set of authors [43]. In author profiling, on the other hand, the goal is to explore some global features that could be used to identify a group of people. This is because of author profiling tasks usually work with texts from a larger size of authors. Thus, the attributes that are found are expected to be more robust, compared to what can be found using author identification [17].

The field of linguistic forensics and text analysis has seen a growth in recent years [42]. The transition from manual author profiling to the use of more sophisticated methods have intrigued many scientists from different areas of expertise. Resulted in a wide range of new techniques that have gathered knowledge from everything from computer science to language studies. Furthermore, the increase of popularity within author profiling can be shown in the growth of the number of participants in different author profiling competitions, like the profiling task at Plagiarism, Authorship and Social Software Misuse (PAN). PAN is a competition where the participants try to determine the age, gender and personal traits (for instance, introvert vs extrovert) based on a given set of data.

There are multiple reasons for the shift in interest that is happening now. One of them is that author profiling has become more useful due to the vast amount of textual data generated each year. For example, the benefit of profiling a suspect based on the textual evidence, thus making the search space for the suspect narrower is undoubtedly an advantage. On the other hand, another reason for the escalation of appeal in this area has to do with an increase of use cases which it can be applied. Over time the use of author profiling has changed from only be used in the forensic investigation and internet security, also to be applied in targeted marketing and advertisement [35]. Companies would, for instance, be interested in obtaining the knowledge of what could describe the people that like or dislike their products [43].

Lastly, the increase in the amount of information itself makes it easier to use known methods to create more accurate results. Moreover, this is also coupled with the ability to better utilise the gathered data due to the rise of computing power.

1.1 Research Question

My research question for the master thesis is:

> *Can you determine the age group of the author analysing the text that he or she writes?*

In this thesis, the experiments that will be performed will be using two age groups. The first age group consist of people below 18 years of age, and the other group of 25 years and above. That is to say, the main goal of this thesis is to explore if it is possible to decide if the author is an adult or a child based on what they are writing.

In order to answer the main research question, several sub-questions needs to be answered as well. For instance, *which language feature or features correlates best with the age of an author?* Every human writes differently, and as we will discuss in a later part of this paper, authors write using different textual features that are unique. On the other hand, even though an author has a unique style, there are similarities across the age groups that can be explored. In the thesis, we will try to examine what kind of writing more likely to be observed at a certain age. Thus, which feature or combination of features that best can determine if the author is a child or an adult.

Another sub-question I will try to investigate is: *can you make a model that is working on many different genres?*. In this thesis, we will be working with texts from different genres. In this case, we will research if the textual feature concerning age works well across all these genres, or is the difference in language substantial enough that different models for each genre are needed.

Further, *which classification algorithm and data set characteristics is significant in regards to obtaining accurate and realistic results?* Throughout the experiments in this thesis, different classification algorithms will be used, and we will examine which characteristics of these algorithms that are most influential to achieve an accurate predicting process. Also, different methods of pre-processing and other factors of the data sets, that could influence the results will be explored.

1.2 Motivation

The main motivational factor in this thesis is to increase internet security. Mostly related to chatroom security and trying to detect fake profiles. For instance, adults posing as a child to get in contact with potential victim children. This could help to determine if a person is whom they pretend to be in the online chatrooms.

Other motivations or use cases for researching within the field of author profiling could be other aspects of internet security. For instance, if an account gets hacked and the hacker post comments in the name of the actual owner. Looking for global features in the text might help to determine that the comments are fake and stop the posts from being posted online. Furthermore, mapping textual features to an age group, gender or other personal traits, can have significant benefits in forensic work. It can help a forensic investigation narrow the potential number of suspects, or even help rule out potential suspects.

1.3 Subject Limitation

The limitation that has been set for this thesis is that the experiment will only determine between two age groups. We will try to tell if the author is below the age of 18 years old, or if they are above the age of 25. There has been earlier research that has been using several age groups, but results in these kinds of studies tend to vary a lot. Pinpointing an age to a small age group is difficult, and is likely the reason for the fluctuating results in the previous research. Furthermore, this is also the reasoning behind having an age gap between the two age groups. Since the main objective is to increase the chat room security, the most crucial task is to distinguish authors between the two age groups. This distinction is to make it easier to decide which age group the author belongs to. This means that another limiting factor in the thesis is that the age group between the age 18 and age 25 will be missed. Further, the thesis will predominately consider the age of the authors. As already mentioned, there are multiple traits like gender, personal traits and origin of the person, that also could contribute to a better understanding of the author. To better focus the experiments in the thesis, we have chosen to only focus on this one trait.

Another limitation I have set to the project is that I will only look at the English language. Furthermore, only use English corpora. Most of the research done in this field is primarily done using English, but there are research done in other languages, and one could also have looked at similarities and differences across different languages.

1.4 Outline

The outline of the thesis is as follows: Starts by exploring the background and theory of the tools utilised in the experiments. For instance, this includes a brief outline of pattern recognition in general, followed by exploring machine learning algorithms and the concept of Natural language processing. In Chapter 3, the state of the art and related works within the field of author profiling will be discussed. A chapter about the data sets follows the State of the art chapter. Which will give a more in-depth analysis of the corpora that were used in some of the previous related work, and which we will be using in the experiments of this thesis. Chapter 5 will discuss the methodology that was used during the thesis. Especially concerns about how the experiments were performed and how the results were gathered. Chapter 6, is where the experiments of the thesis and the corresponding results will be presented. Lastly, the final chapter (Chapter 7) is an overall discussion of the works that were done in this thesis. Furthermore, some concluding remarks as well as some directions of the potential future work.

Chapter 2
Background

2.1 Pattern Classification

Pattern classification, also known as pattern recognition, is specified as methods attempting to automatically distinguish between two or more different instances based on separable patterns [9]. Examples of different instances are human faces, DNA sequences or written texts. Bousquet et al. [9], more formally, described pattern recognition as the task of mapping between the input data X, in order to be able to describe an input pattern, to a class label Y to fulfil $Y = f(X)$. The goal of an accurate pattern recognition algorithm is to produce the smallest possible error rate when mapping f. In other words, the lower the number of mislabelled values of Y, the better the recognition algorithm perform.

When talking about pattern classification, a distinction can be made between two different types of algorithms. The classification types can either be supervised or unsupervised. In supervised classification tasks, the goal is to map an input to an output based on a learning function, that is trained by using example input-output pairs [47]. For this reason, the data is required to be labelled. Unsupervised learning, on the other hand, labels are not included. Thus, the task of this type of algorithm is to find the best partition or clusters of the included data. In this thesis, only supervised learning classification methods will be used when conducting pattern recognition.

Another important requirement of pattern classification is to be able to describe the pattern that the algorithm is dividing its data against. These are called features. For instance, in a text, some features could be the frequency of capital letters and punctuations, or the different topics or the words that are used. Generally speaking, the features can be looked at as the characteristics of the data for a given problem [9].

In the Figure 2.1 we can see a typical pattern classification procedure. It consists

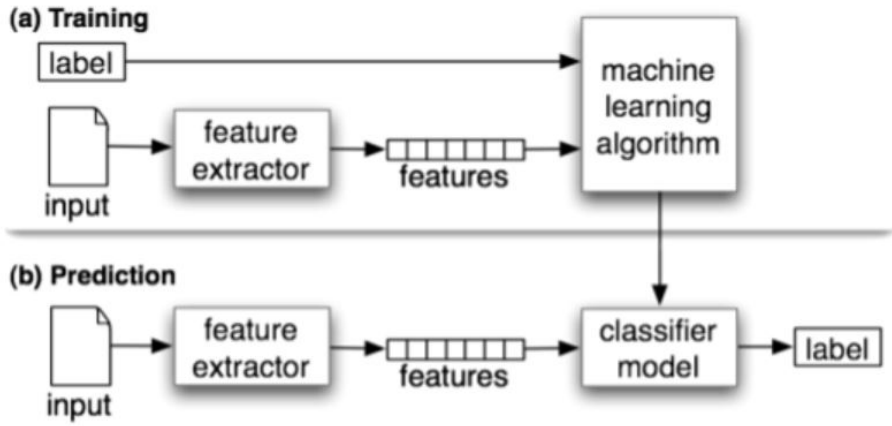


Figure 2.1: A typical procedure of pattern classification. Consisting of two segments, one training component and one prediction sequence [6]. The input in this thesis will be text that will be labelled based on the age of the author. Figure obtained from [6].

of two parts: a) the training sequence and b) the prediction sequence. In the training part, the input or corpus are added with corresponding labels. Then extract the features of the input texts based on the list of features already specified. This list of features with its different weighting, however, is something that would need changes and modifications in order to obtain the most accurate result. Lastly, in the training part, one or more machine learning algorithms are trained on the given list of features, this composes a classifier model. In the second part, new unlabelled inputs are added. Similar to the first part, in the prediction part, the features are extracted based on the same list of features.

Furthermore, based on the classifier model already made, a prediction is performed trying to determine which label the new input has. Lastly, the accuracy of the classifier model is tallied. In the next sections, a more in-depth description of different machine learning models will be presented.

2.2 Support Vector Machine

Support Vector Machine (SVM) is a model used for linear classification and is considered state of the art supervised learning algorithm[33]. SVMs have its theoretical basis from the field of statistical learning theory [55], and is especially suited for binary classification problems. Where the labels usually are classified to the values

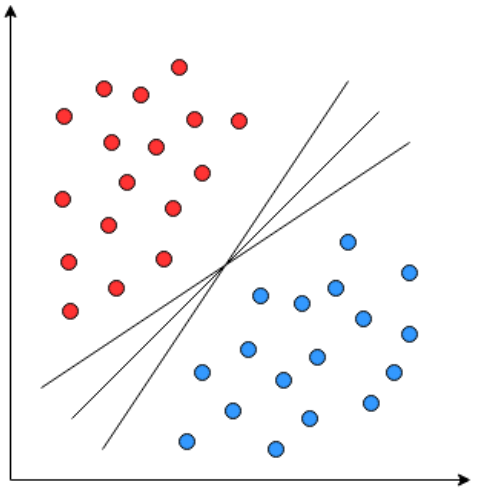


Figure 2.2: Examples of three different hyperplanes.

+1 and −1.

Generally, for a binary classification problem, the SVM has two main tasks to solve:

1. Find a hyperplane within the limits of the input space, that is used to divide the data into two sub-spaces. Examples of different hyperplanes separating the two sub-sets can be seen in Figure 2.2.
2. Maximise the distance from the dividing hyperplane to the border vectors of the two sub-spaces. These border vectors are what is called *support vectors.*

The training data set, in a binary SVM classification problem, is set with input vectors $x = \{x_i\}_{i=0}^{n}$ where $x_i \in \mathbb{R}^N$. This gives that the hyperplane needs to be within the margin of $\mathbb{R}^{N-1}$. The matching labels of x are $y = \{y_i\}_{i=0}^{n}$ where $y_i \in \{+1, -1\}$ [26]. Furthermore, the equation of the hyperplane is defined as:

$$w \cdot x + b = 0 \tag{2.1}$$

Where x is the input vector, w is defining the orientation of the hyperplane and is usually called *weight vector*. Lastly, it is what is called the *bias*, b, which is the value of the offset of the hyperplane in regards to its origin.

In order to satisfy the first of SVMs main task, that a hyperplane should divide the data into two sub-spaces, the hyperplane should ensure that (this assumes

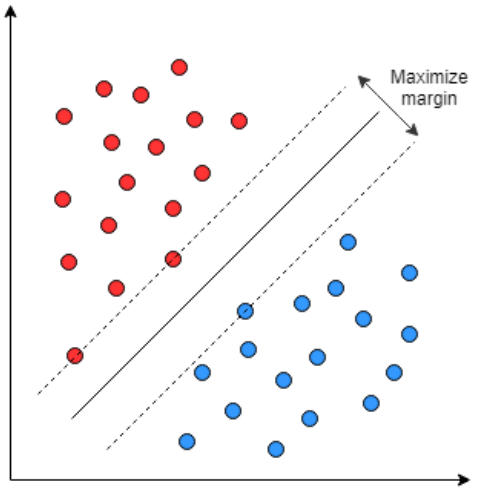


Figure 2.3: Separating hyperplanes. Each of the two new hyperplanes (dotted line) can be described with: $w \cdot x + b = 1$ and $w \cdot x + b = -1$.

$y \in \{+1, -1\}$) [9].

$$y_i \cdot ((w \cdot x_i) + b) > 0 \text{ for all } i = 1, \ldots, m \tag{2.2}$$

In order to calculate the margin between the two subdomains, an additional two hyperplanes are added. These hyperplanes are parallel and share equal offset to the original hyperplane. Similarly, both of these new support hyperplanes will surface the corresponding sub-spaces support vectors. This can be seen in Figure 2.3. Furthermore, combining these two hyperplane equations gives the following general equation:

$$y_i \cdot ((w \cdot x_i) + b) \geq 1 \text{ for all } i = 1, \ldots, m \tag{2.3}$$

Furthermore, the second main problem that SVMs try to solve, which is to maximise the distance to the data set's two sub-spaces. In SVM this is tackled by solving the minimisation problem $||w||^2$, of different dividing hyperplanes. There is only one hyperplane that will realise the maximal distance for a given data set [51, 9]. This can also be written as:

$$\begin{aligned} &\text{minimize} \frac{1}{2}||w||^2 \\ &\text{subject to } y_i \cdot ((w \cdot x_i) + b) \geq 1 \text{ for all } i = 1, \ldots, m \end{aligned} \tag{2.4}$$

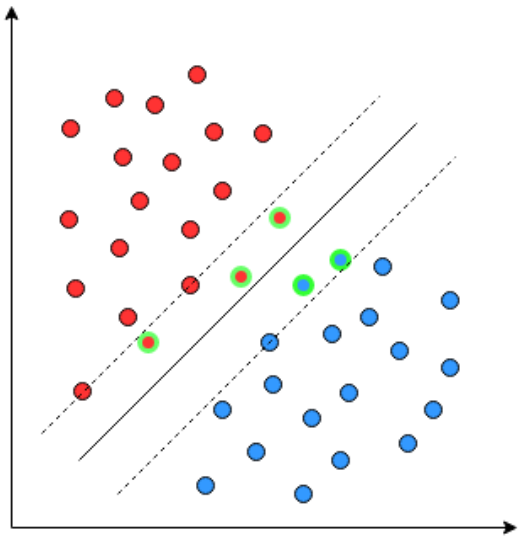


Figure 2.4: Example of soft-margin SVM. The the data points on the 'wrong' side of the support hyperplanes are highlighted in green.

Equation **??** is describing a minimisation problem using, what within SVM, is called hard margin. Hard margin implies that no errors or no noise are tolerated in the calculation. Thus, making the non-linearly separable problems unsolvable. In most cases a *slack variable* is introduced, to lessen the zero noise constraint. This is called a soft margin SVM and is represented as $\xi_1 > 0$ for every input vector x_i. Soft margin allows input data to be placed in the 'wrong' side of the support hyperplanes, as shown in Figure 2.4. If the soft margin SVM is applied to the hyperplane equation 2.3, we get the following equation.

$$y_i \cdot ((w \cdot x_i) + b) \geq 1 - \xi \text{ for all } i = 1, \ldots, m \tag{2.5}$$

Further, adding the a soft margin variable to the minimisation problem in equation 2.4, we get the following equation:

$$\begin{aligned} &\text{minimize } \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i \\ &\text{subject to } y_i \cdot ((w \cdot x_i) + b) \geq 1 - \xi_i \text{ for all } i = 1, \ldots, m \end{aligned} \tag{2.6}$$

In Equation 2.6 a new variable C is introduced. This variable determines the model's complexity and the tolerated distance of the input vectors from the class margin. This trade-off is eased with a lower value of C, and on the other side acts similar to a hard margin SVM when the value of C is high.

Lastly, this constrained minimisation problem can be solved by simplifying the equation using Lagrange multipliers. Furthermore, it can be solved by the use of quadratic programming optimisation algorithms [9].

$$\begin{aligned} \text{Maximise } \alpha, \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_1 \alpha_j y_i y_j (x_i \cdot x_y) \\ \text{subject to } 0 \leq a_i \leq C \text{ and } \sum_{i=0}^{m} a_i y_i = 0 \end{aligned} \tag{2.7}$$

Finally, a classification of new samples of data is done by using the following equation:

$$y = sign(\sum_{i=1}^{m} a_i y_i (x \cdot x_i) + b) \tag{2.8}$$

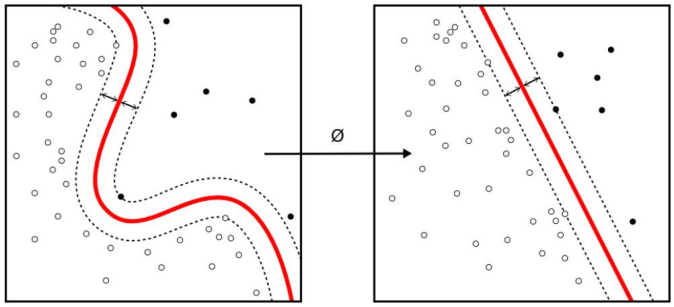


Figure 2.5: The kernel method. Data that is not linearly separable are moved to another dimension where it is easier to divide it. Figure obtained from [14].

When dealing with text classification, or other pattern analysis tasks, often rather than not, the data that needs to be analysed are not easily dividable. In the context of SVM, this means it could be challenging to create a satisfactory hyperplane between the two classes. In machine learning, something called kernels can be used to overcome this issue. The kernel method is based upon transforming data that is not linearly separable, into another, often a higher dimension, where the dividing margin is more distinct [48, p. 690–695]. This can be seen in Figure 2.5.

Two different SVM kernels will be used in the experiment part of this thesis:

- **Linear kernel:** $K(x_i, x_y) = x_i \times x_y$. When the data that is already or close to linearly separable the linear kernel is frequently used.
- **Radial Basis Function (RBF):** $K(x_i, x_y) = \exp(-\gamma||x_i - x_y||^2)$. This kernel method is often preferred since the equation results near to 1 when the values of x_i and x_y are close, and close to value 0 when they are further apart. What could be considered to be close values of x_i and x_y is determined by the γ parameter. With a small value of γ, then the values of x can be further apart to be considered close, and the other side, the values of x_i and x_y needs to be closer if γ is large.

2.3 Naive Bayes

Another supervised learning algorithm used to recognise patterns is the Naive Bayes algorithm. The Naive Bayes classifier is widely used within the field of machine learning due to the algorithm's efficiency and the ability to handle evidence from a large combination of features.

The Naive Bayes algorithm is a probabilistic classifier which is using the Bayes' theorem for applying independent assumptions. As a result of the independent assumptions, the algorithm can be classified as naive [25].

The Naive Bayes classifier consist of two components [15, 31]. Firstly, it is a list of *features* $F_1, \ldots, F_n$ or in this case, text documents. Secondly, A class $C = \{c_1, \ldots, c_m\}$, which denotes the conditional probability of this set of features. By combining these two components together with the general Bayes Theorem, we get the following equation:

$$P(C|F_1, \ldots, F_n) = \frac{P(C)P(F_1, \ldots, F_n|C)}{P(F_1, \ldots, F_n)} \tag{2.9}$$

Further, it is possible to simplify the equation. Because the denominator does not depend on the value of C, it is possible to ignore it altogether. This is possible on the grounds of the *naive* assumption that the features operate independently. Thus, giving the simplified version of the equation:

$$P(C|F_1, \ldots, F_n) \propto P(C) \prod_{i=1}^{n} P(F_i|C) \tag{2.10}$$

Lastly, taking the $argmaxC$ over the different set of $C = \{c_1, \ldots, c_n\}$, will give the probability for the occurrence of a particular class for a given set of features/documents.

Similarly to SVM, Naive Bayes does also have different kernels that could be used. In this thesis, two of these kernels will be utilised:

$$P(F_n|C) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp(-\frac{(F_n - \mu_C)^2}{2\sigma_y^2}) \tag{2.11}$$

Equation 2.11 shows the Gaussian Naive Bayes kernel. This kernel assumes that the likelihood of all the features follows a Gaussian distribution. One can efficiently compute the probability of a feature by using its mean (μ) and standard deviation (σ) values [24]. The second kernel is the Bernoulli kernel. This kernel requires the data to follow a binary classification and assumes to follow a multivariant Bernoulli distribution. The decision rule is based on the following equation:

$$P(F_n|C) = P(n|C)x_n + (1 - P(n|C))(1 - x_n) \tag{2.12}$$

$P(F_n|C)$, in this case, denotes the probability of class C producing the term x_n.

2.4 Challenges concerning Machine Learning Algorithms

One of the main challenges with machine learning algorithms can be classified as a generalisation problem. In other words, how well will a trained machine learning model perform on an unseen and new set of inputs, which may differ slightly from the input the model was trained on [19].

More specifically, when a machine learning model gets trained, we can compute the training accuracy on the training data. Coupled with the training accuracy, we can determine the training error rate, which is, $1 - Accuracy$. The objective of a trained model is to reduce this error rate as much as possible. However, what differentiates machine learning from merely being an optimisation problem is that we also are interested in the test accuracy, or rather the test error rate to be as low as possible as well. This is often called the generalisation error rate and is formally defined as the expected value of the error on new input [19].

Multiple factors can influence the generalisation error rate. However, the main determining factor is to make the training error small, and further make the gap between the training and test error small. When tackling this problem, the concept of machine learning capacity is essential. The machine learning algorithm's capacity is the component of the model that could change and influence the outcome in order to lower the generalisation error rate. By changing the capacity, we aim to manipulate if the model is more likely to overfit or underfit.

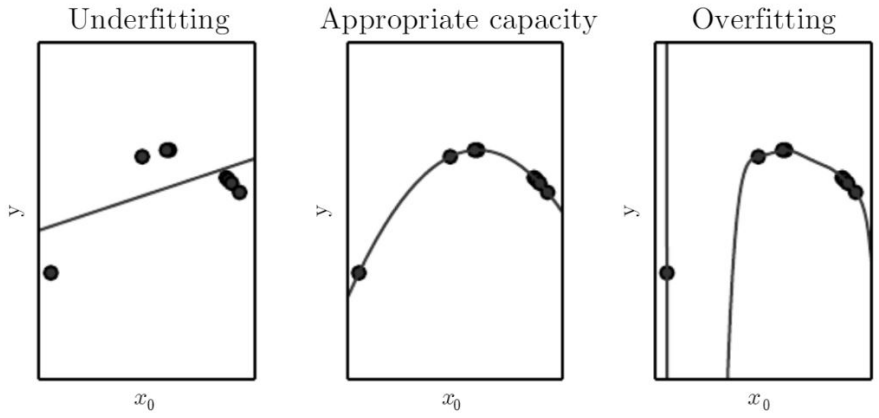


Figure 2.6: Example of underfitting, overfitting and appropriate capacity [19]. When underfitted the model struggles to make a sufficient function to represent the data set. The capacity is appropriate when a generalised function can represent the data in a good manner. The model is overfitted when the function is too explicit regarding a specific data set.

If the model is not able to achieve a tolerable low error rate, the model we have trained is underfitted. This occurs when the the model cannot recognise a sufficient pattern within the training data and struggle to fit the training data into generalised patterns. In this case, we say the model has a low capacity. On the other side, a model with a high capacity is overfitted . Generally, a model is overfitted when the gap between the training and test error rate is to substantial. For instance, if a model picks up the noise or random fluctuations in the training data set and learned as patterns by the model. The model's classifier may be too specific and will produce insufficient results when served unseen input. In Figure 2.6 an illustration of the different concepts are shown.

How to counter this capacity issue and make a generalised model is difficult. One way to reduce the impact of this limitation is to have a sufficient sized training data set. In this way, the model may have adequate data to reduce the training error rate. As well as, have enough data to recognise what part that could be considered irrelevant and what part that could become generalised concepts. However, on the other side, there is a possibility to "over-train" on a data set. That means that in the attempt to reduce the training error as much as possible, one can run training for a long time. Thus, overfitting the model and see an increase in the generalisation error. The relationship between capacity and error can be seen in Figure 2.7. Other means to confront the capacity issue will be discussed in more detail in the methodology

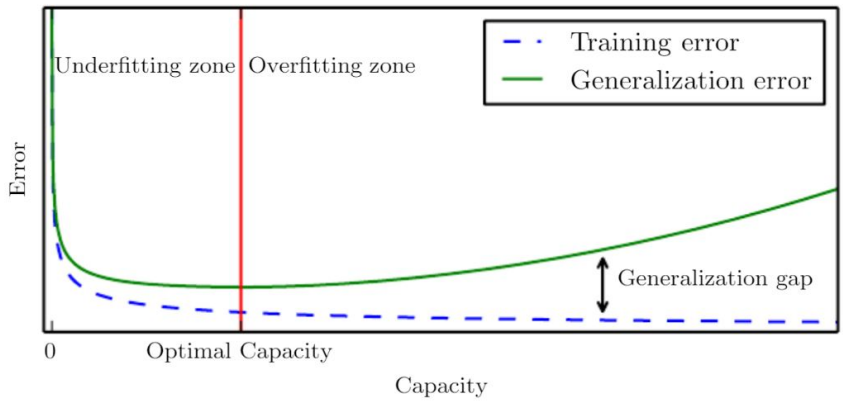


Figure 2.7: A common example of the difference in behaviour between error and capacity. In the underfitting zone, both training error and generalisation error are low. As the capacity increases, both the training error and generalisation error decreases. However, after the optimal capacity of the model is reached the generalisation error starts increasing. Eventually, the gap between the errors outweighs the low training error. Thus we have an overfitted model [19]. Figure obtained from [19].

chapter at a later stage in this thesis.

2.5 Natural Language Processing

With the introduction of Natural Language Processing (NLP) techniques into the field of Author profiling, other ways to achieve the content of a text have been introduced. NLP is a field within computer science, which aims to create methods to read and understand human languages.

One of these approaches is what is known as Part of Speech (PoS) tagging. Firstly, Part of Speech is formal equivalent words that can be collected into classes [12]. Usually, the different classes that exist in the English language are verb, noun, adjective, adverb, pronoun, preposition, conjunction and interjection. Generally speaking, the method of PoS tagging uses probabilistic models to apply the right tags to the words in a text. Some of the main difficulties when using this technique is to classify a word that appears in more than one category. For instance: i) The **run** lasted thirty minutes and ii) We **run** three miles every day [10]. The word "run" is a noun in the first sentence and a verb in the second. This issue is tackled by using a genre-specific and large corpus to train the PoS tagger. Because the tagging accuracy decreases when used on out of domain data [18]. Gimpel et al. [18] made a PoS tagger specialised on the informal language in social media. They trained a

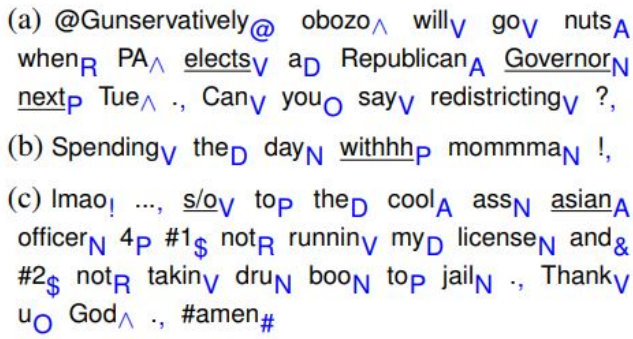


Figure 2.8: Example tweets with PoS tagger annotations[18].

system using a sizeable corpus of twitter messages. This is shown in Figure 2.8.

Another technique used in natural language processing is something called n-grams. This technique falls under what is known as statistical inference. The goal of this approach is to take some data, generated with an unknown probability distribution, and then making some estimation about this distribution [29]. Further, with the n-gram model, the goal is to try to predict the next word. Thus, it can be sated as estimating the probability function of P in from the equation 2.13

$$P(W_n|W_1, \ldots, W_{n-1}) \tag{2.13}$$

Since this is a stochastic problem, the calculation of the most probable next word is based on the classification of the previous words. Thus, in order to have some confidence in the probability of following words of a given classification, much text needs to be analysed. However, in most cases, there will be mostly new sentences that have never been analysed and classified before. In other words, no prior identical textual history that the prediction could be based upon. Moreover, even if the sentence begins according to some recorded sentences seen before, it might have a different ending. One possible way to tackle this issue is to use something called *Markov assumption*. Markov assumption is a method of grouping recorded histories (sentences) that are similar in different ways, in order attempt to give plausible predictions of which words to come. The assumption that is made is that only the last few words have an impact on the next word. An n-gram model is constructed by putting equivalent sentences in the same class if they share the same local context, or rather the same last n-1 words [29].

The most used cases of n-grams are for $n = 2, 3$ and sometimes $n = 4$, and are usually called bigram, trigram and four-gram. In an ideal scenario, we would like

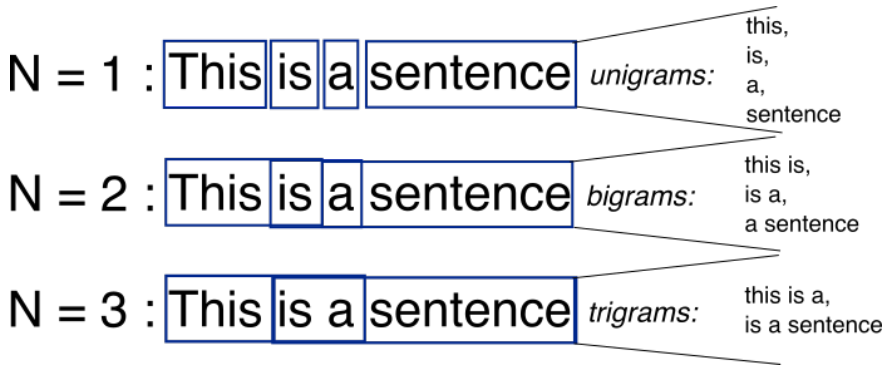


Figure 2.9: Example of how unigrams, bigrams and trigrams work on a simple sentence. Figure obtained from [1].

the value of n in the n-gram model to be large, since a high value of n can cover many edge cases. However, if the data is divided into too many classes, the number of different outcomes increases drastically. Thus, it is too computationally heavy to estimate. Usually, only bigrams and trigrams are deemed practical. An example of how the different results of unigrams, bigrams and trigrams can be seen in Figure 2.9.

Chapter 3
State of the art

3.1 Features

In the field of author profiling, the most common approach has been to perform text classification on the text. The way this is done is to assign predefined class labels to a text. In this case, the main focus of the earlier research has been to find the best resulting textual features. As Ortega-Mendoza et al. [36] points out, there are in particular two kinds categories of textual features that have been playing a central role: Stylistic based- and content based features.

3.1.1 Stylistic Based Features

A stylistic based approach aims to look at the style of the text, or rather how the text was written. Examples of this could be, for instance, the length of sentences, length of paragraphs, how many punctuations/emoticons that are used, the use of capital letters. Furthermore, the use of stop words and function words is also classified within the style of the text. This will be discussed more in depth in Chapter 5.

The most common use case regarding these stylistic textual features is to calculate the frequency that a given feature appears in the text. Furthermore, using different combinations of features is also necessary in order to determine similarities and distinctions in how different age groups use language and how they formulate sentences.

3.1.2 Content Based Features

This text analysis approach, on the other hand, aims to classify the content or context of the text. As already mentioned the main technique in the previous approach is to measure the recurrence of a set of particular stylistic textual features. Whereas in the realm of the content based approaches, that would only be one of the many techniques that could be utilised.

For instance, a common way to achieve insight into the content of the text is to count words over already existing groupings of words. As Schwartz et al. explained in [52], body words like a nose, head, hair, face, can be placed in a body lexicon. Further, every time a word in the analysed texts uses words from the body grouping, it will be counted. Using this data, it can be possible to determine which age group that writes about the body the most.

Within this method of categorising words into different word lexicons, the most used lexicon system is Linguistic Inquiry and WordCount (LIWC) [37, 53, 37], developed by researchers at University of Austin, Texas. The LIWC2015 has over 70 different dictionary lexicons divided into four main categories: i) Summary language Variables, ii) Linguistic Dimensions, iii) Other Grammar, iv) Psychological Processes. Further, there are over 6,400 unique words distributed over the 70 different dictionaries, as well as, many words appear in many dictionaries. For instance, the word *cried* is part of five lexicons: verbs, past focus, sadness, negative emotion and overall effect.

LIWC has been used in a series of studies were the researchers have tried to determine the age and gender of the authors. Some studies have shown that females use more first-person singular pronouns, like "I", "me" and "my", and males use more articles [5, 13]. In regards to age, studies show that older authors tend to use less negative emotions and less use of first person singular pronouns [13, 38].

Another method that can be utilised to classify the content of a text is n-grams. As mentioned in chapter 2.5, n-grams looks at different ways to split a sentence in order to understand the context and topics of the given sentence better. Since different age groups and genders often speak about different topics. Using n-grams, with different values of n, it is possible to calculate the different frequencies of topics mentioned by the different author groups. For instance, looking at blogs on the internet, topics such as football, computer and car tend to more frequent in blogs written by male authors. On the other side, words like shopping and husband will increase the probability that the blog has a female author. By analysing many texts, the different N-gram frequencies of different topics written by male and female authors can be calculated. Further, using the words with the most distinct ratios can be used as features [49].

However, using, for instance, only unigrams or bigrams can misrepresent the context/content of the sentence. Having a sentence like: "I hate shopping", can produce different results based on the value of n. Using unigram, which only looks at one word at the time, will most likely conclude that this sentence has a female writer. Because, when comparing "I" and "hate" independently, the frequency is not distinct enough to say whether the author is female or male. Unlike, "shopping", as

already mentioned, tends to be more frequent as a topic in female blogs. On the other hand, using trigrams, the conclusion will most likely be shifted more towards a male writer. Since the whole sentence has been included, and the context of the whole sentence is taken into account.

3.2 Earlier Work

As mentioned earlier, there has been a growing interest in research within the field of author profiling. Furthermore, the most accurate results have been achieved by using combinations of features from both the content based- and the stylistic based approaches. Schler et al. [50] looked at the effect of writing styles in blogging with the regards of gender and three different age groups. The age groups were divided into teens (13-17), young adults (23-27) and adults (33-47). They collected a corpus containing over 71,000 blog posts and looked at several different textual features, with emphasis on function words, hyperlinks and non-dictionary words (e.g. slang words). They achieved determining the gender of the authors with an 80% accuracy and the age group of the authors with a 75% accuracy. In particular, the result showed a correlation between the age groups and their use of prepositions and determiners. The result obtained by Schler et al. was further improved by Goswami et al. [20]. By adopting similar techniques to a 20,000 large blog corpus, they increased the accuracy to 89.2% in gender identification and 80.3% in determining the author's age group. They found equivalence between the use of particular slang words and the average length of sentences used in the blogs, with the age and gender of the authors.

With the rising popularity and prestige concerning the Authoring Profiling task at the PAN events, new insight is obtained. At PAN in 2013, the task was to identify the age and gender from a large social media corpus. Most of 18 participants used combinations of different stylistic features, such as frequency of capital letters, quotations, punctuations and emoticons. [43, 39]. As well as the use of POS- tags and HTML specific traits, like image URLs and web page URLs.

Furthermore, the content based features used by the participants were mainly Latent Semantic Analysis, TF-IDF, dictionary/topic based classifiers such as LIWC and bag of words. The classifying approaches used by the participants were all supervised machine learning techniques. Most of the participants used decision trees, support vector machines and logistic regression. Meina et al. [11] obtained the highest accuracy in the competition, with a 59.2% gender accuracy and 64.9% age accuracy. It was achieved using linear SVM classifier. Furthermore, using features such as PoS-tagging, n-gram, counting the intensity of particular words, and the frequency of errors and abbreviations.

Another author profiling competition was conducted at PAN in 2014. Similarly to the objective of the 2013 competition, the goal of PAN 2014 was to obtain the age and gender of the authors. Unlike the previous PAN event, the corpus of PAN 2014 is more varied and consists of a combination of blogs, hotel reviews, social media and Twitter posts, both in English and Spanish. Similar classification methods and content- and style- based features that were used in PAN 2013, were also utilised in the 2014 competition [42]. Further, the highest values of accuracy were obtained by Maharjan et al. [27] with 73.4% in gender identifications, with English Twitter messages, and 61.1% in age identifications with Spanish Twitter messages. They used models with different combinations of character- and word-based n-grams. Building several models for each of the four corpora categories, as well as building a joint model that would combine all the different genre. For the sake of investigating what could be different genre-specific traits versus more generalise textual features. However, the average result in the age classification was somewhat lower than the previous year. The main reason for this was most likely the more fine-grained age group that was introduced.

In contrast to the three different age groups in 2013 (10s, 20s and 30s), in 2014 the number of age groups of the authors that needed its own label, was increased to five (16-24, 25-34, 35-49, 50 and 65+). Additionally, there were no gaps between the age groups, like it was in the classification of the 2013 corpus. This increases the difficulty to create apparent distinguishing traits between each age classification.

3.3 Limitations and weaknesses with the current approaches

One of the major issues of using these kinds of approaches is that the result of the study seems very dependant on the context of the corpus. The result of similarly used methods varies a lot based on different genres, and there have been studies that conclude with contradicting results. For instance, studies [32, 45] have concluded that females tend to use emoticons more often, than males. While another study [52], concluded with the polar opposite. The reason for this is most likely the difference in context or genres of the corpus, used by the two studies. Further, it can be hard for researchers to determine how generalised the result of these studies are, what applies to, for instance, age or gender, or just applicable to the corpus.

In other words, one could claim that this weakness of contradicting results stems from the difference in language over different genres. Trying to determine what correlates between age and the corpus or just corpus specific features is difficult. As already mentioned, this issue has been addressed with the corpus of four different genres in the PAN 2014 competition as well as other researchers. Nguyen et al. [34] tried to tackle this area using a joint model on three different types of genres. They obtained an accuracy of 74% trying to determine the age of the author based on

features found in all the different parts of the corpora. The accuracy obtained is not terrible, but is somewhat lower than research done within a narrower scope of the genre.

Furthermore, another of the limiting factor researchers have to undertake when researching within this area, it that the data size has to be substantial for the methods used to be effective. This is not an issue that only applies to this author profiling, but an essential factor when working with this type of machine learning algorithms as the results see an increase in accuracy in correlation with a larger sized corpus. This results in that most of the studies on this topic need to have a reasonably large sample size of authors. As has to be noted, the need for a large corpus also gives a basis for another common problem in author profiling studies, which is to gather the necessary data about the authors efficiently. Sometimes do the researches does not have the necessary data of the authors, which means the researchers need to label the data manually [42].

Chapter 4
Data Set

4.1 Schler Data Set

The Schler data set consist of blogs from over 71 000 authors from blogger.com. All the blogs were gathered from the blog site in August 2004, and they downloaded only blogs with self-provided gender indication. Further, the corpus consists of 681 288 different blog posts. All of them with the length of at least 500 words in total. Of the minimum 500 words, there are at least 200 occurrences of common English words.

As shown in Figure 4.1, a little under 25 000 authors have an unknown age. These

	gender		
age	**female**	**male**	**Total**
unknown	12287	12259	24546
13-17	6949	4120	11069
18-22	7393	7690	15083
23-27	4043	6062	10105
28-32	1686	3057	4743
33-37	860	1827	2687
38-42	374	819	1193
43-48	263	584	847
>48	314	906	1220
Total	34169	37324	71493

Figure 4.1: The distribution of gender and age in the Schler corpus [50].

Table 4.1: The distribution of age in the PAN 2013 corpus [49].

	10s	20s	30s	Total
Male	8 600	42 900	66 800	118 300
Female	8 600	42 900	66 800	118 300
Total	17 200	85 800	133 600	236 600

will be filtered out because it will not be possible to label these authors correctly. Further, the authors in between the age of 18 to 25 will also be filtered out, since this age group falls between the two age groups this project. This results in approximately 11 000 authors within the first age group (13-17 years) and 15 500 authors that are classified as 25 years and above.

4.2 PAN 2013 Data Set

The PAN 2013 corpus consists of a large set of blogs. The corpus has an equal number of blog posts per gender. However, it is fairly uneven in regards to age. As indicated in Table 4.1, there are only 17 200 authors that are classified as teens (13-17). On the other side, there are 219 400 blogs combined from authors in the 20s (23-27) and 30s (33-47).

The blogs were collected from several blogging sites, such as netblog.com and blogspot.com, as well as collected from different themes of blogs. Resulting in a diverse range of topics, which aims to make the profiling task more realistic. This also provides the opportunity to explore standard cliches, either reinforcing or disproving them. For instance, younger people talks more about the school, homework and video games and older people talks more about news and work. Additionally, another attempt to make the framework classification of the corpus more realistic is to include both long and short blog posts.

In contrast to the Schler data set, the PAN 2013 corpus does not have the same lower word count limit. Figure 4.2 shows the distribution of the total numbers of words per blog post in the corpus. The same figure also displays that the average blog post consists of 335 words.

4.3 PAN 2014 Data Set

As mentioned in Chapter 3.2, the corpus of PAN 2014 was of a more varied nature. It consisted of four different genres: blogs, hotel reviews, social media texts and Twitter posts. Further, as previously discussed, the number of age groups for labelling was

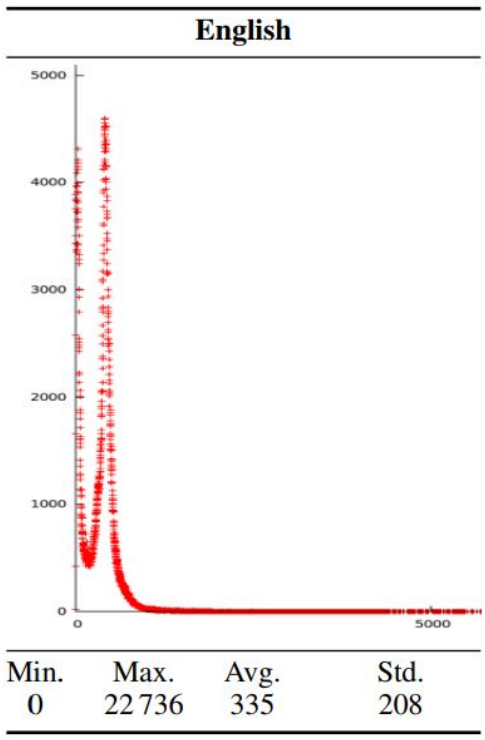


Figure 4.2: The distribution of words per blog post in the PAN 2013 corpus [43].

increased in the 2014 corpus, from three to five age groups. The distribution of the different genres in respect of the five age groups is shown in Table 4.2 below. In regards to the main objective of this thesis, the hotel review genre of the corpus has been deemed not relevant, and it will not be included. Mainly because of the more formal nature of the reviews included in the corpus, in comparison to the more informal language of the three other genres.

Social Media: This corpus part consists of entries from the PAN 2013 data set. It was selected from authors who had an average number of words per blog post greater than 100 words.

Blogs: The blog part of the 2014 corpus is the biggest of the four genres and the objective from the PAN staff when collecting the blogs was to make the gold standard for author profiling in the blog genre [42]. With this intention, the blog entries were manually selected. As well as, verified manually that the blog was written in English and updated by one person. For each author in the corpus, it is included a maximum of 25 blog posts.

Table 4.2: Distribution of Blogs, Social Media and Twitter authors with respect of age classes [42].

	Blogs	Social Media	Twitter
16-24	2370	20	34
25-34	1080	90	150
35-49	3426	68	204
50-64	2788	37	90
65+	52	8	12
Total	9716	223	355

Twitter: In the same way as the blog genre, the Twitter users with the corresponding tweets, were manually included. Different Twitter users from several occupations (eg. journalist and teacher) were chosen, as well as different levels of opinion based Twitter users (Influencers vs Non-influencers), to attempt to give a realistic representation of the twitter users. For each author in the corpus, it is included a maximum of 1000 tweets.

4.4 PAN 2015 Data Set

The data set used in the 3rd edition of the author profiling competition at PAN 2015, consist of Twitter users with corresponding Tweets. Similarly to the Twitter corpus from the 2014 PAN competition, the users were selected from a variety of occupations, age groups and levels of opinion based Twitter users. As shown in Figure 4.3, the age groups used are also similar to the 2014 edition. The only difference is the 50-64 and 65+ from 2014, are combined into a 50+ age group. For this thesis, this data set only provides text for the adult group (25 years and above).

Table 4.3: Twitter user distribution with respect of age classes [44]

	Twitter Users
16-24	130
25-34	134
35-49	48
50+	24
Total	336

Chapter 5
Methodology

5.1 Implementation

The approach in author profiling task is usually divided into several tasks. Mainly these tasks are, firstly, formalise the data set gathered. Further, perform feature extraction and implement one or more classifiers. The process of this thesis can be divided into four steps, which will be discussed more in depth in the following subsections.

5.1.1 Data Set Preparation and Formalisation

In the first step, the goal is to prepare the data set for feature extraction and further training of the classifier. Firstly, the data sets needed to be acquired. This was done using [40] for the data set gathered from the PAN competitions and [22] for the Schler blog corpus.

The next step in the formalisation process is to gather all the texts from the different data sets that suited the thesis problem description, as well as pre-processing of the selected data. The pre-processing process is important because it generalises the data from all the different corpora, so there will be no obvious biases in the classification process. For instance, the different texts are represented in the same format, which exclude the format in itself as a feature for the classifier. Additionally, in the pre-processing, potential noise in the data set can be removed, which can yield more accurate classifiers. Several pre-processing techniques were utilised on the data set:

- **Tokenizing:** Firstly, tokenize all the sentences in the data set. This removes all unwanted white spaces and makes it easier to run for instance, PoS- tagging and n-grams techniques in the training and testing phase. Due to the informal nature of the data set in this thesis, a twitter tokenizer provided by Natural Language Toolkit (NLTK) was used [2]. The reason behind this, is that this

tokenizer also works well with special characters as smiles, hashtags etc. Further, it preforms well on non-twitter texts as well. An example of the tokenizing process is: "This is a cooool #dummysmiley: :-) :-P <3 and some arrows < > -> <–", that becomes, "['This', 'is', 'a', 'cooool', '#dummysmiley', ':', ':-)', ':-P', '<3', 'and', 'some', 'arrows', '<', '>', '->', '<–']"

- **Remove URLs and HTML tags:** Since the data set were provided in the .XML format, a lot of HTML- tags were still present in the text. The tags such as "
", "<a>" etc. were removed. For the URL strings, on the other hand, they were removed and replaced with "url". This is because an author's use of URLs could become a useful feature, but the contents of the URL in itself is not important. The same goes for image links in the text, these links were replaced by "image".

- **Stop words:** A stopping words can be defined as a commonly used word in a given language. These words often do not carry much meaning, but only serve a syntactic function [16]. In English stopping words can be "a", "an", "the", "in", "on" etc. Stop words can have a different impact on the accuracy of the result. Firstly, since they tend to have a high frequency, stop words often diminish the impact of other less common words. Which again can influence the importance of these words. By removing the stop words, there will be an increase in the relative frequency of the "non-stop words". Secondly, removing the stop words can increase the processing speed, since it reduces the number of tokens the system needs to store [28]. For this thesis, the stopping words list provided by NLTK will be used [2]. Note that in this thesis, there will be made separate data sets with and without stop words, to investigate the impact these may have.

- **Stemming:** Stemming is the process of limiting the forms a word can be used in a given text, to a base form. Words like 'is, are, am' becomes the joint word 'be'. For instance, 'the boy's cars are different colours' becomes 'the boy car be differ colour' [28]. Stemming can be a useful method to reduce the number of different features, and can also make the data set less 'noisy'. Similar to the stop words, there will be made a separate data set where stemming is taken into account. In order to explore the potential impact it may have on the different classifiers.

Removing URLs, removing stop words and stemming of the data set can also be helpful in order to mitigate both overfitting and underfitting of the machine learning algorithms. Because it will remove parts of the text that may get the classifier to evaluate words or text features as concepts.

5.1.2 Feature implementation and Engineering

The next step in the process is to implement and extract features from the data set. In the list below is a summary of the different features that was used in the training and test phases in during the experiments. The features selected are based on the state of the art research done in the field of author profiling and more general information retrieval research.

1. **Term frequency-inverse document frequency (TF-IDF):** This is a common technique in the field of author profiling. The way TF-IDF works is by determining the frequency of a word in a text in the corpus and compare it to the inverse frequency of the word over the whole corpus [41]. The TF-IDF score for a given term t can be represented by the equations below:

$$TF(t) = \frac{\text{Number of times t appears in a document}}{\text{Total number of terms in the document}} \tag{5.1}$$

$$IDF(t) = \log(\frac{\text{Total number of documents}}{\text{Number of documents with t included}}) \tag{5.2}$$

$$\text{Term score(t)} = TF(t) \times IDF(t) \tag{5.3}$$

   Terms that are common like stop words will get a low relative score compared to more rare ones. It is intended to present the relative importance of a given term. Furthermore, there will be an equal amount of word features as unique words in the corpus.

2. **N-grams:** As discussed in Chapter 2.5, looking at different n-grams can be beneficial in determining the different age groups. In the different experiments, I will be looking at the frequencies of different unigrams, bigrams and trigrams that appear in the corpus. Then further map the different n-grams to the different age groups. I will conduct different experiments using both character-based n-grams and word-based n-grams.

3. **Part of Speech:** Also discussed in Chapter 2.5, Part of Speech tagging is a useful method in the realm of author profiling. In this thesis, the PoS-tagging library provided by Textblob [4] were used. The way this was measured, was to calculate the frequency of a given PoS-tag in the text, and further compare the difference in frequency between the two age groups. Table 5.1 shows the different PoS-tags that were used.

Table 5.1: Part of Speech- tags that was used in the training and testing phase during the experiments [3].

Tags	Description	Example
NN	Noun, Singular	chair, tiger
NNS	Noun, Plural	chairs, tigers
PRP	Pronoun, Personal	me, you, it
PRP$	Pronoun, Possesive	my, your, our
WP	Wh-pronoun, Personal	what, who, whom
WP$	Wh-pronoun, Possesive	whose, whosever
VB$	Verb, base form	think
VBZ	verb, 3rd person singular present	she `thinks`
VBP	verb, non-3rd person singular present	I `think`
VBD	Verb, past tense	they `thought`
JJ	adjective	nice, easy
JJR	adjective, comparative	nicer, easier
JJS	adjective, superlative	nicest, easiest
RB	adverb	extremely, hard
WRB	wh-adverb	where, when
IN	conjunction, preposition	of, on, before, unless
CC	conjunction, coordinating	and, or, but
DT	determiner	the, a, these

4. **Stylistic feature frequencies:** Further, as well as the frequency of Part of Speech tags, stylistic features frequencies from the texts, are measured. These features include:
   - Word count, the total number of words in a text
   - Short word count, the total number of short words less than three characters.
   - Unique word count ratio, unique words divided on the total number of words in a text.
   - Character count, the total amount of characters used in a text.
   - Average word length, the average length of the words used in a text.
   - Punctuation count, the number of punctuation used in a text.
   - Emojis count, the number of used emojis in a text.
   - Upper case count, the number of upper case letters in a text.
5. **Function words:** The LIWC word list has been used to significant effect in previous author profiling studies. We will investigate which of the 70 subdictionaries of the LIWC that works best for the different genres in the data set in this thesis.
6. **Latent Dirichlet Allocation (LDA):** LDA is often used to detect the underlying topic in a given text document. It works with the assumption that text with similar topics will use complementary groupings of words [8]. In the context of this thesis, LDA will be used in order to find relations between topics from different texts.

5.1.3 Training the model

The third part of the process is to train a classifier or several classifiers on a labelled data set, using the extracted features from the previous step. Different machine learning classifiers or models come with different characteristics and properties. Running multiple models can of this reason be beneficial. Because, after evaluating the result of the different models, it is possible to find out which model performs best on a given corpus.

The first of the classifiers that are used is the Naive Bayes method (Chapter 2.3). As discussed in the State of the Art Chapter 3, Naive Bayes is a commonly used method in previous research within author profiling, as well as frequently used by the participants in the PAN competitions. One of the strengths of the Naive Bayes classifier is that it is easy to set up and fast in the prediction phase. Besides that, it is a generally robust classifier that produces accurate results [30]. This means that it

could be considered for a real-time prediction system. Due to these characteristics of Naive Bayes, the results will work as a baseline for other classifier results to compare itself against.

On the other side, however, the downside of using Naive Bayes is when the assumption of independence between the features does not hold. This could lead to high fluctuation in the classifier prediction result when running multiple times, even with the same configurations on the same corpus.

The other method that will be used in this thesis is the Support Vector Machine classifier. SVM is one of the most common supervised learning method used in the field of author profiling. The reasoning behind this is because SVM can deal with a high amount of features, which already discussed, is the case in author profiling tasks, due to the often large corpus size. Further, the SVM classifier is also used because of its robustness to overfitting, in that the algorithm uses effective feature selection to diminish irrelevant features [23]. However, compared to Naive Bayes and other simpler classifier algorithms, the time SVM uses to train on the corpus is noticeably higher.

Lastly, another aspect regarding these kinds of supervised learning classifiers is hyperparameters. In contrast to feature values, hyperparameters are values that are set before the training of the classifiers begins, not derived from the training itself.

Due to its simple nature, the Naive Bayes classifier does not provide the option of adjustable hyperparameters. However, we will use the two different Naive Bayes kernels, the Gaussian and Bernoulli kernels as discussed in Section 2.3, as means of investigating predictions of the Naive Bayes classifiers.

In regards to SVM, more hyperparameters can be adjusted. Also, in SVM, the two different kernels discussed in Section 2.2, the linear and radial basis function kernel, will be applied. When using the linear kernel, one of the hyperparameters that are adjusted is the value of C, which modifies the tolerated distance of the input vectors from the class margin. [21] states that the most common value of C is found in the $2^{-}5, 2^{15}$ range. This range also holds when using the rbf kernel. Furthermore, besides changing the value C in the rbf kernel, the value of γ is also changed. The most used values of γ are commonly in the range of $2^{-15}, 2^{3}$.

Lastly, hyperparameters to the TF-IDF method can also be adjusted. Especially concerning the frequency values of terms. It can be beneficial to remove both words with the highest and/or lowest word frequency value. For the sake of reducing the number of word features that the classifier needs to process, thus reducing the training speed.

5.1.4 Re-train and validate

The final step in this process is to improve the performance of the classifier, by changing and altering the different features, as well as the hyperparameters mentioned above. The order this was done in the experiments, was first to modify and alter the combination of features extracted from the data set. These features were discussed in section 5.1.2. When an adequate result is achieved with different mixes of features, further tuning of hyperparameters were done on the best achieving combination of features.

The approach that was used to find the best achieving mix of features was made using a simple A and B test. This means only changing one feature at the time and comparing it to the most accurate result. The benefits of using this method are that it gives a more methodical insight on the impact of each single features. However, this process is slow, and due to time limitation, it is not possible to cover all of the different combinations.

Table 5.2: An example of a rbf cross-validation sequence. It consists of the number of the test as well as suggested values of C and γ

	C	γ
1	2^{-5}	2^{-15}
2	2^{-3}	2^{-13}
3	2^{-1}	2^{-11}
	$\vdots$	
10	2^{15}	2^{3}

Additionally, before testing on a corpus, it is not possible to know which value of C (SVM linear kernel) or combination of C and γ (SVM rbf kernel), that is best fitted for the task. In order to identify the best alternatives of these values, $v - folded$ cross-validation was used. In this method, we divide the training set into v subset of the same size (in the case of this thesis, $v = 5$). Next, one subset is tested on the training set combined of the other subsets. Training is done with $v - 1$ sets and testing with the remaining set and repeated where all the sets are once used for testing. For each new training phase, trying the values of C and the pair of C,γ. In the case of the rbf kernel, C.W. Hsu et. al. [21] propose a grid search approach. This means that changing the value pair of C and γ in an exponential sequence yielded good results. An example of a cross-validation sequence is shown in Table 5.2.

5.2 Result evaluation

In order to give a more accurate evaluation of the results of the classification performance, a confusion matrix is utilised. The main reason behind this is that using the accuracy of the classification, which is the number of correctly classified objects divided by the total number of objects [48, p. 8], as the only metric when evaluating the result can lead to inaccurate conclusions. For instance, the accuracy measure could be exceptionally susceptible for unbalances in the data set (if there is a lot more of one of the two classes).

The confusion matrix is a two-dimensional matrix that includes the original class label on one dimension and the assigned class on the other dimension. In this assignment, a binary version of the confusion matrix is used. This is a particular case of the confutation matrix, having one of the two classes designated to a positive class and the other class described as negative. Figure 5.1 shows the different possible outcomes of the two classes.

		Assigned class	
		Positive	**Negative**
Actual class			
	Positive	TP	FN
	Negative	FP	TN

Figure 5.1: A binary confusion matrix, with the possible outcomes of positive and negative classes [48].

- **TP (True Positive)**: A given sample classifies as positive and is also in the actual positive class.
- **FP (False Positive)**: A given sample classifies as positive but is in the actual negative class.
- **FN (False Negative)**: A given sample classifies as negative but is in the actual positive class
- **TN (True Negative)**: A given sample classifies as negative and is also in the actual negative class.

Furthermore, with the use of the confusion matrix, it is possible to derive different metrics, which make it possible to give a more thorough performance assessment. These metrics are: recall, precision, F_1-score as well as the accuracy.

– **Accuracy:** As already mentioned, accuracy is the number of all correctly classified classes divided by the total number of classes. This is the most common performance metric used [7]. Moreover, it can be described by the following equation:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.4}$$

– **Precision:** precision is a measure of the percentage of samples that the classifier labels as relevant that is actually relevant. In other words, it is the total number of samples retrieved that are relevant, divided by the total number of documents that are retrieved [48, p. 990-991] [54]. The following equation can also describe the precision of class c:

$$Precision_c = \frac{TP}{TP + FP} \tag{5.5}$$

– **Recall:** Similarly to precision, is recall also a measure of how the classification algorithm retrieves the relevant information. The difference between the two is that recall describes the total number of documents retrieved that are relevant, divided by the total number of relevant documents in the corpus [48, p. 990-991]. The following equation can represent recall of class c:

$$Recall_c = \frac{TP}{TP + FN} \tag{5.6}$$

– F_1**-Score:** This is a binary classification specific measure, that evaluates the prediction of classification. It works as a harmonic mean between the precision and recall measures, this means the F_1- value lies between the two other metrics values, but slightly closer to the lowest of the two [48, p. 497]. The F_1- measures can be represented by the following equation:

$$F_1 - measure = 2 \times \frac{precision_{avg} \times recall_{avg}}{precision_{avg} + recall_{avg}} \tag{5.7}$$

Chapter 6
Experiments

In this chapter, we will go through the four experiments that were performed for this thesis. We will talk about the experiment's objectives as well as the results and the general observations done.

6.1 Experiment 1: Initial testing

The goal of the experiment was to implement a working program that would use some of the methods used in previous research. It was conducted using the PAN 2013 blog data set. Additionally, it would also provide some benchmark results of the training, which could be expected to be outperformed in the later experiments.

The feature that was used in the initial phase consisted off most of the simple frequency calculations. For instance, word count, short word count, character count, average word length, punctuation count and upper case count was used.

As we can see from Table 6.1, the SVM with linear kernel produced the best results in almost every measure. It was done with a C value of 2^1. The SVM with RBF kernel had C value of 2^3 and γ of 2^{-3}. However, as we can see from the results

Table 6.1: Result of the initial test. It includes the accuracy, precision, recall and F_1 score, using the two classifiers, Naive Bayes (NB) and Support Vector Machine (SVM), with different kernels. This was trained on the PAN 2013 blog corpus.

Classifier	Accuracy	Precision	Recall	F_1
NB, Gaussian	0.51	0.62	0.52	0.57
NB, Bernoulli	0.46	0.46	0.52	0.49
SVM, Linear	0.55	0.56	0.60	0.58
SVM, RBF	0.52	0.30	0.63	0.41

that the highest accuracy is 55%, which is substantially lower than the prediction obtained by the PAN 2013 competitors. Due to the fact that the task is to distinguish the text from two different age groups, did the three best scoring classifiers just hardly beat a 50/50 random prediction. Whereas, the Naive Bayes with Bernoulli kernel did perform with an accuracy lower than random guesses.

6.2 Experiment 2: Testing on the different genres

In the next experiment, we conducted testing on the different genres individually. The main objective was to improve the prediction score from the initial test. As well as, try to explore which feature and hyperparameter combination that performed best on the three genres that the whole corpus consists of (blogs, social media, and Twitter). The way this was approached, was first to conduct training on the blog data set. Attempting to increase the accuracy on that subcorpus. Further, using the best performing model on the other genres to explore how accurate a model performs on another domain. Furthermore, tuning the features and hyperparameters to, if possible, enhance the accuracy of prediction of the other genres classifiers.

6.2.1 Blogs

This genre does consist of the largest percent of the whole data set. As previously mentioned in chapter 4, the Schler data set, the PAN 2013 data set (this was used in the initial test) as well as a subset of the PAN 2014 corpus, all consist of blog posts.

Table 6.2: Best result of training on the Schler data set. SVM Linear had C value of 2^5, SVM RBF had C value of 2^4 and γ of 2^{-5}.

Classifier	Accuracy	Precision	Recall	F_1
NB, Gaussian	0.82	0.89	0.80	0.84
NB, Bernoulli	0.79	0.77	0.77	0.77
SVM, Linear	0.93	0.87	0.92	0.90
SVM, RBF	0.88	0.91	0.88	0.89

We can see from the Tables 6.2, 6.3 and 6.4 the results were similar. Furthermore, we also observe that SVM with the linear kernel did have the most accurate prediction overall. With the highest scoring accuracy of 92% on the PAN 2013 data set. As well as a 94% precision score, a 91% recall score and a 92% F_1 score. Also, the hyperparameters were similar over the training of the three different blog data sets. In the case of the linear SVM, the best performing value of C was around 2^5 to 2^6. For the SVM with the RBF kernel, the best combination was with C of $2^3 to 2^4$ and γ equal 2^{-5}. If we compare the results from Table 6.1 and Table 6.3, the prediction

Table 6.3: Best result of training on the PAN 2013 data set. SVM Linear had C value of 2^6, SVM RBF had C value of 2^3 and γ of 2^{-5}.

Classifier	Accuracy	Precision	Recall	F_1
NB, Gaussian	0.84	0.90	0.79	0.84
NB, Bernoulli	0.78	0.65	0.63	0.63
SVM, Linear	0.92	0.94	0.91	0.92
SVM, RBF	0.88	0.89	0.86	0.87

Table 6.4: Best result of training on the PAN 2014 blog data set. SVM Linear had C value of 2^6, SVM RBF had C value of 2^3 and γ of 2^{-5}.

Classifier	Accuracy	Precision	Recall	F_1
NB, Gaussian	0.87	0.87	0.89	0.87
NB, Bernoulli	0.78	0.69	0.59	0.60
SVM, Linear	0.90	0.89	0.89	0.89
SVM, RBF	0.91	0.89	0.91	0.90

accuracy is significantly increased from the initial test to the testing done on the same data set.

The feature combination that was utilised in all of the blog experiments was TF-IDF where the 5% of the most common words and 1% of the least common words not taken into account. Both word level and character level bigrams and trigrams were used. Part of Speech tagging with the adjective tags (VB$, VBZ, VBP and VBD), pronoun tags (PRP, PRP$, WP and WP$) and noun tags (NN and NNS). As well as LDA and all of the stylistic frequency features listed in section 5.1.2.

With these features in mind, some observation that was made in during this experiment was: TF-IDF with 5% of the most common not taken into account had the best result of all the during the experiments on the blog corpus. Although, most of the words that were cut out of the TF-IDF's process were stop words. However, it did perform better using TF-IDF than the tests where the stop words were removed in the pre-processing stage. This may be explained by the fact that when using stop words list, it does not adapt to the specific corpus in the same way TF-IDF does. By removing the 5% most frequent words from the calculation, it is more able to reduce the noise in the corpus than a stop word list may be. Another argument that goes against the use of pre-processing of stop words is that many of the stop words are pronouns, like 'me', 'my', which was a very useful metric. We observed that the

Table 6.5: Best result of training on the combined data set of Schler, PAN 2013 and PAN 2014 blog data set. SVM Linear had C value of 2^5, SVM RBF had C value of 2^5 and γ of 2^{-5}.

Classifier	Accuracy	Precision	Recall	F_1
NB, Gaussian	0.84	0.87	0.85	0.86
NB, Bernoulli	0.80	0.95	0.67	0.79
SVM, Linear	0.90	0.88	0.87	0.87
SVM, RBF	0.89	0.85	0.91	0.87

text written by children had a more significant frequency of pronoun PoS-tags than adults.

It was also observed that the adult authors had a generally higher PoS frequency of adjectives in their texts. Coupled with adults having observably more adjectives than children in the blog genre, the frequency of adjectives was a useful feature in the blog genre as a whole. In contrast to, for instance, Twitter posts, where the number of adjectives was lower, and it was not as good of a feature. Furthermore, we observed that adults used longer words on average than child authors, as well as texts written by adults had a higher unique word ratio than children. Given these points, it may be possible to conclude that adult authors in this corpus, often write with a more extensive sized vocabulary. Moreover, the use of punctuation was also a good feature to distinguish the two age groups. Children authors used, on average more punctuation than adults, and especially the use of '!' was more prevalent by the younger authors.

From Table 6.5 we see the result from the experiment with the data set consisting of all of the three blog data sets combined. Correspondingly with the trend from the training on each of the individual blog data sets, the SVM with linear kernel did preformed the best, in almost every metric. Further, we also observe that the Naive Bayes with Bernoulli kernel had its best performance in the joint blog experiment, with its highest accuracy of 80 % and highest precision score 95%. However, it had a rather low recall score of 67%. This tells us that the classifier predicted a high amount of children authors that were in fact children, but did also label a fair amount of authors as adults, when they were in fact children.

6.2.2 Social Media

Due to the similarities of the social media corpus with the 2013 PAN blog corpus, the same feature combination that was used in the blog experiment did also performed well on the social media corpus. As seen in Table 6.6, the results of the best

Table 6.6: Best result of training on the PAN 2014 social media data set. SVM Linear had C value of 2^4, SVM RBF had C value of 2^5 and γ of 2^{-3}.

Classifier	Accuracy	Precision	Recall	F_1
NB, Gaussian	0.88	0.93	0.89	0.89
NB, Bernoulli	0.80	0.69	0.67	0.67
SVM, Linear	0.91	0.90	0.89	0.89
SVM, RBF	0.91	0.89	0.91	0.90

performing social media experiment were similar to the results of the blog corpus. We can observe that SVM with the RBF kernel did achieve the highest accuracy as well as the best F_1 score. Furthermore, the only difference between the model used in the blog experiment and the one used in the social media experiment was some slight changes in the classifiers hyperparameters. The best performing social media experiment had a C value of 2^{-3} with the linear kernel, and a C, γ combination of $2^5, 2^{-3}$ with the RBF kernel.

Although there were many similarities to the feature composition used in the social media corpus compared to the blog corpus, there were also some minor differences. For instance, the LIWC dictionaries concerning informal language and swearing words had a higher frequency and worked well as a distinguishing feature. Also, words concerning work or school was a good indicator of the age group of the author in this corpus.

6.2.3 Twitter

The last of the sub-experiments on the individual data sets consisted of training on the Twitter corpus. This corpus includes the Twitter part of the PAN 2014 data set, as well as the PAN 2015 data set. In contrast to the blog corpus experiment 6.2.1, where all the different parts of the blog corpus also were tested individually, this was not done with this experiment. Due to the fact, the Twitter data sets were individually much smaller.

Furthermore, as mentioned earlier, the best scoring feature combination on the blog corpus was tested on the Twitter corpus. The results from that analysis can be seen in Table 6.7. We can see that the prediction done using the blog experiment's feature combination did not achieve to score any high prediction values in the different metrics.

However, from Table 6.8, the feature mix is changed, and the classifiers perform more accurate on the same corpus. Again, the best performing classifier is the linear

Table 6.7: The result of training on the Twitter data set, with the features from the combined blog experiment. SVM Linear had C value of 2^{-3}, SVM RBF had C value of 2^{1} and γ of 2^{-7}.

Classifier	Accuracy	Precision	Recall	F_1
NB, Gaussian	0.65	0.70	0.55	0.57
NB, Bernoulli	0.64	0.69	0.55	0.57
SVM, Linear	0.77	0.87	0.76	0.81
SVM, RBF	0.73	0.76	0.80	0.77

Table 6.8: Best result of training on the PAN 2015 Twitter data set. SVM Linear had C value of 2^{6}, SVM RBF had C value of 2^{5} and γ of 2^{-5}.

Classifier	Accuracy	Precision	Recall	F_1
NB, Gaussian	0.86	0.91	0.85	0.88
NB, Bernoulli	0.81	0.94	0.76	0.84
SVM, Linear	0.87	0.92	0.89	0.89
SVM, RBF	0.86	0.87	0.76	0.81

kernel SVM, with almost best scores in every metric. The features that were changed was among other things, the word and character level bigrams and trigrams, to word and character level unigrams and bigrams. This may be because of the size limitation of the Tweets. In Twitter, there are only 140 characters in a Tweet, which makes that the authors need to be more delicate in his or her wordings. Using unigrams worked better than trigrams in this corpus, which may be due to the unigrams effect of distinguishing the difference in the use of abbreviations. Words like 'omw' (on my way), 'lol' (laugh out loud), was more frequent in the younger demographic.

Also, some Part of Speech tags were not as effective with the Twitter data set as with the blogs, for instance, the adjective tags (JJ, JJR and JJS). As mentioned before, the adjective tags had a better effect on the blog corpus, this could be explained by the blog texts often is more descriptive as a genre compared to Twitter. Tweets may not be as descriptive due to the character limitation. Another feature that was not as helpful in Twitter posts, where the use of punctuation. The general use of punctuation was much lower compared to in blog posts. This made it more challenging to use punctuation as a distinguishing feature.

On the other side, some features that had observed effect on the Twitter corpus was, for instance, the Part of Speech tags of verbs (VB$, VBZ, VBP and VBD)

and nouns (NN and NNS). Also, the upper case count worked better on the Twitter corpus, than on the blog data set.

TF-IDF was also utilised in this experiment. Unlike the blog experiment, where 5% of the most common words and 1% of the least common words were not taken into account, this did not yield the same good result for the Twitter corpus. Using TF-IDF on the Twitter data set, worked best with discarding the 1% most common words and, similarly, the 1% of the most uncommon words. The reason for this is perhaps the character limit on Tweets. As already discussed, the limitation makes that the author needs to be more careful of his or her wordings. Which makes that the authors might not include all the common stop words or write in complete sentences. Then, there is not as much noise in the data set.

6.3 Experiment 3: Different age groups

Although, the main objective in this thesis is to distinguish authors from two age groups stated in the introduction, which are children, authors of the age 18 and below, and adults, authors of age 25 and above. However, it could be enlightening to see how the differences in language changes from authors of a smaller subset compared to authors below the age of 18. In other words, how authors of the age group of, for instance, age 20 to 29, age 30 to 39, age 40 and above compare with authors below 18. By looking at smaller age groups, a pattern might be revealed of the way language is used and evolves as the authors get older. Which further can be used to understand the textual differences between children and adults better.

The goal of this experiment is to examine the main textual traits of smaller age groups of adults compared to children. We have used the text from the blog data sets and made a balanced data set of an equal amount of entries from each of the different age group.

Table 6.9: Best result of the blog corpus with authors from age group 13-18 against age group 20-29. SVM Linear had C value of 2^{-1}, SVM RBF had C value of 2^{1} and γ of 2^{-7}.

Classifier	Accuracy	Precision	Recall	F_1
NB, Gaussian	0.73	0.88	0.66	0.74
NB, Bernoulli	0.71	0.80	0.72	0.73
SVM, Linear	0.77	0.90	0.71	0.79
SVM, RBF	0.69	0.73	0.70	0.70

Table 6.10: Best result of the blog corpus with authors from age group 13-18 against age group 30-39. SVM Linear had C value of 2^6, SVM RBF had C value of 2^4 and γ of 2^{-3}.

Classifier	Accuracy	Precision	Recall	F_1
NB, Gaussian	0.85	0.90	0.82	0.84
NB, Bernoulli	0.80	0.72	0.70	0.70
SVM, Linear	0.89	0.92	0.85	0.89
SVM, RBF	0.82	0.88	0.85	0.86

Table 6.11: Best result of the blog corpus with authors from age group 13-18 against age group 40 and above. SVM Linear had C value of 2^6, SVM RBF had C value of 2^6 and γ of 2^{-3}.

Classifier	Accuracy	Precision	Recall	F_1
NB, Gaussian	0.91	0.91	0.88	0.90
NB, Bernoulli	0.91	0.88	0.87	0.87
SVM, Linear	0.95	0.96	0.90	0.91
SVM, RBF	0.93	0.92	0.91	0.91

From the Tables 6.9, 6.10 and 6.11 we can see the best resulting scores from the different age groups of adults against children age group. Again did the SVM with linear kernel perform the best in each of the sub-experiments. However, more importantly, we can see the change in performance as the gap between the age groups increases. Especially does the difference between the children and adults above 40 years old become clear. As the best performing classifier have an accuracy of 95%, which are the best result in all of the experiments that we have done.

The features used under this experiment are similar to the one used in the blog experiment. Moreover, another observation that was made during the testing was that the best scoring feature combination was also similar throughout all the different age groups. It was mostly the importance of some of the features that changed. This means that the difference between the written texts was easier to distinguish as the age gap between the authors increased. Firstly, the frequency of the use of punctuation decreased as the age of the author increased. Secondly, as seen in the blog experiment, the number of unique words were also increasing as the age of the author increased.

Some of the other features that were used in this experiment did not have the

same pattern as the ones just mentioned. Although the features were effective to distinguish the children authors from the other writers, but there was not as much of an evolution in the feature's effect as the age gap was increased. Examples of these features were the Part of Speech tags. Since we conducted this experiment on the blog corpus, the same tags that were effective on the blog experiment were also working in this experiment. We also ran tests with the other tags from Table 5.1, as verbs, determiners and conjunctions, but it did not have a noticeable difference in the outcome. Further, different combinations of TF-IDF frequencies were also tried. In other words, we tried the effect of changing the TF-IDF upper and lower thresholds. However, the best resulting thresholds were similar to the ones used in Section 6.2.1.

6.4 Experiment 4: A joint model of all the genres

The last experiment that was executed in this thesis was a joint corpus experiment. In other words, have a classifier train on a data set consisting of the text from all the different genres.

The best scoring result can be seen in Table 6.12. In this experiment, we used similar features that were used with the blog corpus. The hyperparameters in this experiment were similar as well.

Table 6.12: Best result of training on the joint corpus. SVM Linear had C value of 2^5, SVM RBF had C value of 2^9 and γ of 2^{-3}.

Classifier	Accuracy	Precision	Recall	F_1
NB, Gaussian	0.82	0.85	0.80	0.81
NB, Bernoulli	0.79	0.77	0.77	0.77
SVM, Linear	0.89	0.92	0.78	0.84
SVM, RBF	0.88	0.86	0.92	0.89

Due to the imbalance of blog text compared to the two other genres, the features that perform well on the blog corpus, do also achieve good results on the joint corpus. In order to reduce this imbalance, we made a more balanced corpus were all the three genres have equal amounts of entries. The result of this experiment can be seen in Table 6.13. As we can see, the classification prediction done on the balanced has decreased by quite a margin. Notably, the difference in the linear kernel SVM prediction result between the two data sets is around 14% in the accuracy metric.

From the observations done in experiment two, we get a similar outcome in the joint data set experiment. In that, especially the Twitter data set and the blog

Table 6.13: Best result of training on the joint balanced corpus. SVM Linear had C value of 2^{-1}, SVM RBF had C value of 2^4 and γ of 2^{-5}.

Classifier	Accuracy	Precision	Recall	F_1
NB, Gaussian	0.70	0.89	0.62	0.69
NB, Bernoulli	0.68	0.77	0.69	0.70
SVM, Linear	0.75	0.86	0.70	0.74
SVM, RBF	0.75	0.83	0.81	0.81

corpus have some fundamental genre differences. This makes it difficult to create combinations of features general enough to work effectively on both domains. The most effective method in the joint data set was to include most of the different feature combinations from the blog, social media, as well as the Twitter experiments. For instance, in the case of n-grams, both unigrams, bigrams and trigrams on a character and word level were included. The same goes for Part of Speech tags, were adjectives, pronouns and nouns that worked well with blogs and social media were added. As well as, the PoS of verbs that worked well on the Twitter corpus. However, some of the frequency features gave misleading results. As mentioned earlier, the punctuation count was not as useful in the Twitter experiment as in the blog experiment. This was also the case for the word count, short word count and character word count. These features were not as effective in this experiment, thus not included.

There were also some features that had a more noticeable effect on the joint corpus, than on the other experiments. This includes the LIWC dictionary regarding personal concerns (Work, achievement, leisure, home and work) and the dictionaries containing assent (words like agree, Ok, yes) and fillers ('you know', 'I mean'). Additionally, the use of LDA increased the prediction score with the joint corpus.

Chapter 7
Discussion and Conclusion

7.1 Discussion

As we can observe from the experiments done, the classifier that had the overall best performance was the SVM with the linear kernel. This means that the data points in the data sets are already linearly separable. Moreover, it is not necessary to transform the features into other dimensions to become linearly separable. However, the next best performing classifier overall was the RBF kernel-based SVM classifier.

Further, we can observe that the performance of both the Naive Bayes classifiers generally did worse than SVM. This might be due to the number of features that the Naive Bayes classifier needs to handle, and the assumption of independent features is not met. This means that the features calculated by the Navie Bayes classifier are somewhat correlated. Something that could have increased the prediction results for the Naive Bayes is to make feature reduction. This means that the redundant features, with a high correlation value with other features, would be removed. Another point is that Naive Bayes, compared to SVM, does not need the same amount of data to create sufficient prediction patterns. In other words, the vast amount of data used in the experiments may have been too much for the Naive Bayes classifier to 'handle'.

On the same topic of Naive Bayes classifiers, from the results, the Bernoulli kernel does have overall inferior results than the Gaussian kernel. The Bernoulli kernel tends to achieve lower accuracy results compared to the Gaussian kernel when the feature space is large [46]. Despite this fact, due to time limitations, we did not perform specific testing with feature reduction and feature weighing that could probably increase the Bernoulli kernel's results.

From the results of the experiments, we can see that the precision measure is in most cases, higher than the recall value. This means that the classifiers are generally predicting text from, for instance, child authors, where they are in fact children. However, the generally lower recall score over the different experiments, tells that

the classifier also misses some of the relevant documents. Since in the experiments, we want to distinguish children authors from adult authors, the trend of a generally lower recall score, can be a sign of a too 'picky' classifier.

Throughout the different experiments, the hyperparameters from the two SVM classifiers also follows a noticeable trend. In the experiments where the features are more linearly separable the value of C tents to be higher. In the blog experiment (Section 6.2.1) the SVM linear kernel value of C is 2^5 to 2^6, which are on the higher end of the spectrum. This means that the classifier does not allow many outlier data points. In other words, with a high value of C, the classifier is behaving more like a harder margin SVM which penalises misclassification more severely. On the other side, the experiments where there the feature set was not as suitable for its domain, the value needed to be lower. This is the case for the results in the Tables 6.7, 6.9 and 6.13. Whereas the value of C differ from 2^{-1} to 2^{-3}. This means that the cost of misclassification is significantly reduced and the classifier allowed more outliers in the prediction phase.

In the case of the hyperparameter of the SVM with RBF kernel, we can see a similar pattern as with the linear kernel, concerning the C value. In the experiments where the feature set was more appropriate to the corpus, the value of C was relatively high (2^3 to 2^9). Further, in the experiments where the combination of features was not as suitable, the value tends to be lower (2^{-1} to 2^{-3}). Similarly, as the case for the linear kernel, the reason of this might be that the features in the transformed space are not as linearly separable when the feature set used is not fitted towards the genre. The best achieving values of γ was $2^{-3}, 2^{-5}$ and 2^{-7}. This means that the SVM decision boundary is influenced by features that could be distant from the decision boundary. However, the γ value, on the other hand, did not have a clear trend. In that, there was no clear pattern between a low value of C and a low value of γ and vice versa.

As already discussed in the Experiments chapter, different feature combinations were effective in different genres. This is mostly due to the difference in writing style between the domains. Although some features were useful on the different domains. The use of TF-IDF worked well with different combinations of upper and lower thresholds. However, which upper and lower limit was not consistent over the different experiments. Thus it was not possible to observe a general trend.

Furthermore, different types of n-grams were used in the experiments. The experiments concerning the blog and social media data sets, bigrams and trigrams were most effective. The Twitter corpus had best results with unigrams and bigrams. Another, trend that was observed was that the use of both word n-grams and character n-grams produced the best results.

The use of different LIWC dictionaries worked best on the joint corpus experiments. Especially the list containing work/school-, achievement- and leisure-words. The reason LIWC did not amount to practical classifications in the other experiments is still somewhat uncertain. It could, of course, be as simple as LIWC dictionaries did not amount to any classification pattern.

Removing stop words were not necessary for any of the different experiments. Mostly due to the importance of many stop words, especially in the experiments concerning the blog data sets, as it served as a distinguishing feature. Furthermore, the test on the data sets where stemming of the words was taken into account, did not yield as good results as not doing it. Comparing the results of the experiments done in this thesis and the earlier works done on the same data sets shows an increase in accuracy. For instance, Goswami et al. achieved an 80.3% accuracy of the age on the Schler data set, whereas the best-resulting model from the experiments in achieved a 93% accuracy. Similarly, with the works on the PAN data sets, there is a significant increase in accuracy. Such as the PAN 2013 competition, where the best score was 64.9%. In the experiments done in this thesis, 92% prediction accuracy was reached. However, a direct comparison between the different earlier works and the results in this thesis is not possible. This is mainly due to the difference in scope between the works. In this thesis, we have looked at the author profiling classification as a binary classification problem, in comparison to the more divided age groups of both the work on the Schler and the PAN data sets. As well as, there exists an age gap between the age groups in the data set used in this thesis.

7.1.1 Experiments limitation

One limitation of the experiments done is that the corpses used were not balanced around age. Most of the data sets were often balanced in regards to gender, or not balanced at all. The way this was combated in the experiments was to balance the data sets manually, but this also meant that the data set needed to be reduced. In the last experiment, this is shown, where the amount of blog texts heavily outweighs the other two genres. The results of that experiment are of this reason artificially skewed. In the second table (Table 6.13) the results are more realistic since the data set has been balanced with an equal amount of entries of each genre.

Another weakness in the experiments is the similarities between the used social media corpus and the blog data set. The texts used in the social media corpus is, as mentioned in Chapter 4, gathered from the PAN 2013 blog data set. Although the texts were handpicked due to its resemblance to 'real' social media texts, it does, however, not avoid the fact that it is based on the same texts. This results in that the feature combination of the social media experiments is substantially overlapping the blog experiment's feature set. Which also makes the results of the social media

experiment the least reliable of the experiments performed.

As mentioned in Section 6.3, the experiment concerning different age groups was only done using the blog data set. This decision was based both on the time constraint of the thesis as well on the fact that the blog corpus was the most extensive corpus and was the corpus we had worked with the most. Giving the best possibility of obtaining notable results.

7.2 Conclusion

The main goal of this thesis was to investigate if you can determine the age group of an author by analysing the text that he or she writes. The age groups I was mainly concerned about was authors of the age below 18 and the author above the age of 25. Based on the experiments done in this thesis, it is possible to determine the age group of an author. However, many factors are important for the prediction process to be accurate.

Firstly, which kind of textual features that obtained the best results were genre specific. This means that the model trained on other genres, then it was tested on performed poorly, in opposition to if the model was trained and tested on the same genre. Based on the experiments done in this thesis, it was challenging to make a general model that worked well across all the different genres. Ultimately, making individual models for each genre obtained greater accuracy. Furthermore, some of the textual features that were useful in determining the age groups across the different genres were, for instance, the term frequencies using TF-IDF, some specific dictionaries in LIWC, n-grams and stylistic frequencies as Part of Speech tags.

Additionally, the experiments aimed to investigate which classification algorithm that would be most accurate in the classification process. In the thesis, we used two types of Naive Bayes classifiers, using Gaussian kernel and Bernoulli kernel. As well as, two types of SVM classifiers, using a linear kernel and RBF kernel. Based on the results of the experiments, the overall best-resulting classifier was the linear SVM.

Lastly, we wanted to investigate the impact of the results in regards to the different characteristics of the data sets. From the experiments, we have that doing extensive pre-processing of the data set did not enhance the result. Similarly, the fact that the corpora used needed to be as balanced as possible was also significant.

7.3 Future Works

There are different directions for future works that can be done. One possible approach that can be investigated in regards to the difference in language across

the genres is using a two-stage model. In the experiments done, it was only done using texts knowing which genre the text belonged to. Having a two-stage model could make the classifier models more adaptable. By having in the first stage a classifier that is trained for trying to identify the genre of a given text it is most likely contained in. In the second stage, a classifier tries to predict the age group of the author, based on the genre prediction model from the first stage. By using this type of two-stage approach, one could also include other genres than the ones used in this thesis.

Another extension to the implementation of a classifier is to use an artificial neural network (ANN) approach in addition to the supervised learning algorithm used in this thesis. One of the drawbacks of using algorithms like SVM and Naive Bayes is once a classifier model is trained, this classifier will be used on all the given texts. If the classifier needs to be changed or enhanced, then a the training procedure needs to be done again. Using an ANN approach, the classifier will continuously try to improve itself as long as new text is added. If a commercial author profiling software would be made, an ANN approach would probably be suiting.

In addition, in this theses, the experiments done have only included two age groups. Another logical direction is to extend the number of age groups or look at different age groups altogether. One could also try to predict the age more exact, for instance, by utilising binary search. Where the first step could be first to investigate if an author is above or below the age of 50. If the author is predicted below 50 years old, then the target age is changed, and an examination of the age of the author is above or below the age of 25, and so on. Together with a more precise age prediction, one can also look at other describing traits in combination with age. As mentioned in the introduction, other traits could be gender, place of origin or personal traits. Lastly, in this thesis, we have only been working with texts in English. Expanding the experiments to including other languages would be a natural step.

References

[1] Capstone report stephanie. https://rstudio-pubs-static.s3.amazonaws.com/94682_5a41796c9e1a4345b6ba72275612768c.html. Retrieved: 05.06.19.

[2] Nltk tokenize package. https://www.nltk.org/api/nltk.tokenize.html. Retrieved: 07.05.19.

[3] Penn treebank ii tag set. https://www.clips.uantwerpen.be/pages/mbsp-tags. Retrieved: 08.05.19.

[4] Textblob: Simplified text processing. https://textblob.readthedocs.io/en/dev/. Retrieved: 08.05.19.

[5] S. Argamon, M. Koppel, J. W. Pennebaker, and J. Schler. Mining the blogosphere: Age, gender and the varieties of self-expression. *First Monday*, 12(9), 2007.

[6] S. Bansal. Ultimate guide to understand implement natural language processing. https://www.analyticsvidhya.com/blog/2017/01/ultimate-guide-to-understand-implement-natural-language-processing-codes-in-python/, 2017. Retrieved: 29.04.19.

[7] G. E. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[9] O. Bousquet, U. von Luxburg, and G. Rätsch. *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*, volume 3176. Springer, 2011.

[10] E. Brill. A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing*, pages 152–155. Association for Computational Linguistics, 1992.

[11] C. M. BrodziŤska, Celmer, M. Patera, J. Pezacki, and M. Wilk. Ensemble-based classification for author profiling using various features. 2013.

[12] R. W. Brown. Linguistic determinism and the part of speech. *The Journal of Abnormal and Social Psychology*, 55(1):1, 1957.

[13] C. Chung and J. W. Pennebaker. The psychological functions of function words. *Social communication*, 1:343–359, 2007.

[14] J. Dhindsa. *Generalized Methods for User-Centered Brain-Computer Interfacing*. PhD thesis, 2017.

[15] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, 2012.

[16] I. A. El-Khair. Effects of stop words elimination for arabic information retrieval: a comparative study. *International Journal of Computing & Information Sciences*, 4(3):119–133, 2006.

[17] D. Estival, T. Gaustad, S. B. Pham, W. Radford, and B. Hutchinson. Author profiling for english emails. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pages 263–272, 2007.

[18] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. Technical report, Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science, 2010.

[19] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[20] S. Goswami, S. Sarkar, and M. Rustagi. Stylometric analysis of bloggers' age and gender. In *Third International AAAI Conference on Weblogs and Social Media*, 2009.

[21] C.-W. Hsu, C.-C. Chang, C.-J. Lin, et al. A practical guide to support vector classification. 2003.

[22] S. A. J. Schler, M. Koppel and J. Pennebaker. The blog authorship corpus. http://u.cs.biu.ac.il/ koppel/BlogCorpus.htm. Retrieved: 15.02.19.

[23] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[24] G. H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 338–345. Morgan Kaufmann Publishers Inc., 1995.

[25] J. Lin. *Automatic author profiling of online chat logs*. PhD thesis, Monterey, California. Naval Postgraduate School, 2007.

[26] E. Lundeqvist and M. Svensson. Author profiling: A machinelearning approach towards detecting gender, age and native language of users in social media, 2017.

[27] S. Maharjan, P. Shrestha, and T. Solorio. A simple approach to author profiling in mapreduce. In *CLEF (Working Notes)*, pages 1121–1128, 2014.

[28] C. Manning, P. Raghavan, and H. Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.

[29] C. D. Manning, C. D. Manning, and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[30] M. Martinez-Arroyo and L. E. Sucar. Learning an optimal naive bayes classifier. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 1236–1239. IEEE, 2006.

[31] A. McCallum, K. Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

[32] M. L. Newman, C. J. Groom, L. D. Handelman, and J. W. Pennebaker. Gender differences in language use: An analysis of 14,000 text samples. *Discourse Processes*, 45(3):211–236, 2008.

[33] A. Ng. Cs229 lecture notes. *CS229 Lecture notes*, 1(1):1–3, 2000.

[34] D. Nguyen, N. A. Smith, and C. P. Rosé. 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*, pages 115–123. Association for Computational Linguistics, 2011.

[35] M. B. op Vollenbroek, T. Carlotto, T. Kreutz, M. Medvedeva, C. Pool, J. Bjerva, H. Haagsma, and M. Nissim. Gronup: Groningen user profiling. *Notebook for PAN at CLEF*, 2016.

[36] R. M. Ortega-Mendoza, A. P. López-Monroy, A. Franco-Arcega, and M. Montes-y Gómez. Emphasizing personal information for author profiling: New approaches for term selection and weighting. *Knowledge-Based Systems*, 145:169–181, 2018.

[37] J. W. Pennebaker, R. L. Boyd, K. Jordan, and K. Blackburn. The development and psychometric properties of liwc2015. Technical report, 2015.

[38] J. W. Pennebaker and L. D. Stone. Words of wisdom: Language use over the life span. *Journal of personality and social psychology*, 85(2):291, 2003.

[39] M. Potthast, T. Gollub, F. Rangel, P. Rosso, E. Stamatatos, and B. Stein. Improving the reproducibility of pan's shared tasks. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 268–299. Springer, 2014.

[40] M. Potthast, B. Stein, P. Rosso, and E. Stamatatos. Pan: Evaluation data. https://pan.webis.de/data.html. Retrieved: 15.02.19.

[41] J. Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142. Piscataway, NJ, 2003.

[42] F. Rangel, P. Rosso, I. Chugur, M. Potthast, M. Trenkmann, B. Stein, B. Verhoeven, and W. Daelemans. Overview of the 2nd author profiling task at pan 2014. In *CLEF 2014 Evaluation Labs and Workshop Working Notes Papers, Sheffield, UK, 2014*, pages 1–30, 2014.

[43] F. Rangel, P. Rosso, M. Koppel, E. Stamatatos, and G. Inches. Overview of the author profiling task at pan 2013. In *CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, pages 352–365. CELCT, 2013.

[44] F. M. Rangel Pardo, F. Celli, P. Rosso, M. Potthast, B. Stein, and W. Daelemans. Overview of the 3rd author profiling task at pan 2015. In *CLEF 2015 Evaluation Labs and Workshop Working Notes Papers*, pages 1–8, 2015.

[45] D. Rao, D. Yarowsky, A. Shreevats, and M. Gupta. Classifying latent user attributes in twitter. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, pages 37–44. ACM, 2010.

[46] S. Raschka. Naive bayes and text classification i-introduction and theory. *arXiv preprint arXiv:1410.5329*, 2014.

[47] S. J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[48] C. Sammut and G. I. Webb. *Encyclopedia of machine learning and data mining*. Springer Publishing Company, Incorporated, 2017.

[49] K. Santosh, R. Bansal, M. Shekhar, and V. Varma. Author profiling: Predicting age and gender from blogs. *Notebook for PAN at CLEF*, 2013, 2013.

[50] J. Schler, M. Koppel, S. Argamon, and J. W. Pennebaker. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205, 2006.

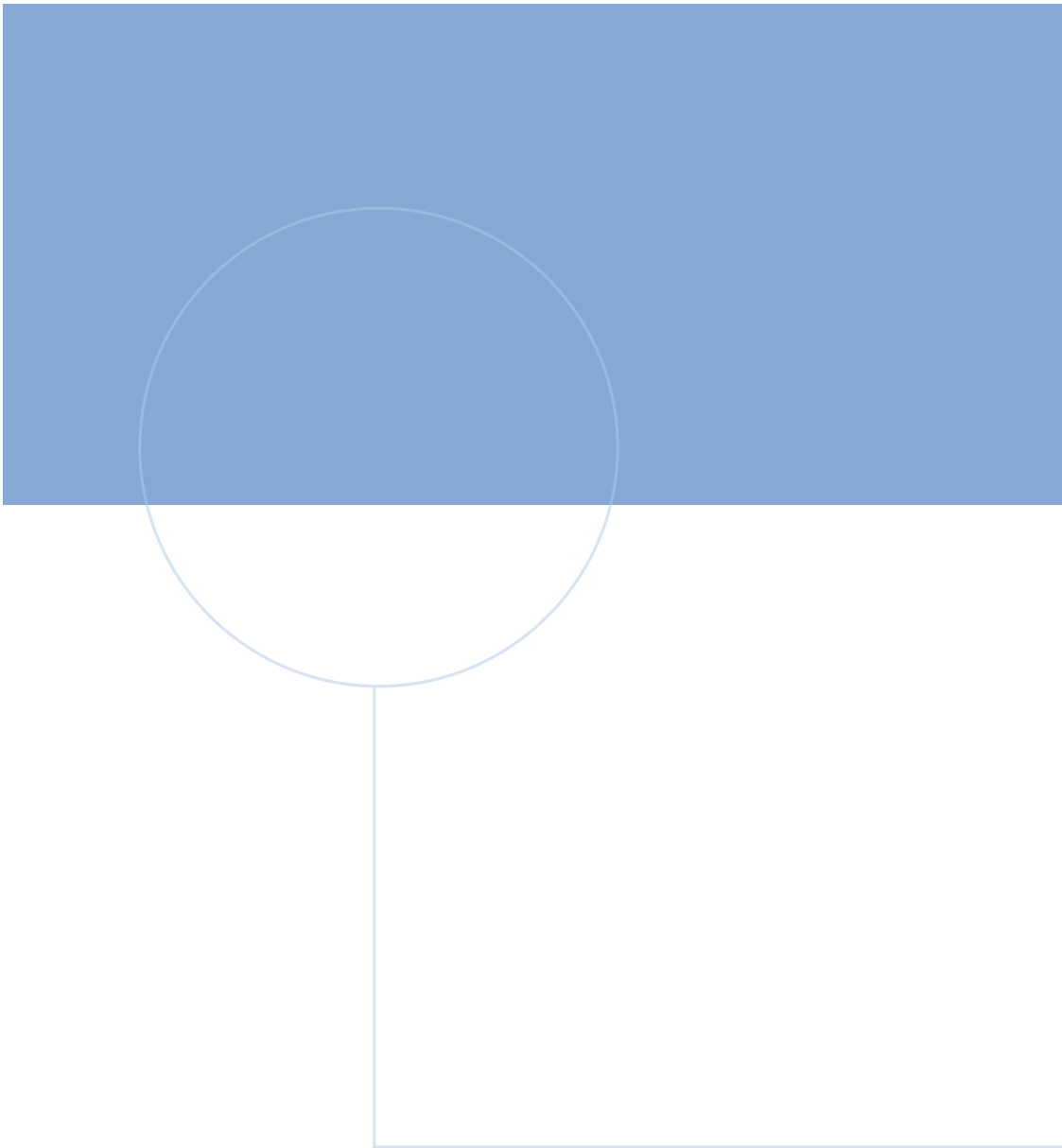
[51] B. Scholkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[52] H. A. Schwartz, J. C. Eichstaedt, M. L. Kern, L. Dziurzynski, S. M. Ramones, M. Agrawal, A. Shah, M. Kosinski, D. Stillwell, M. E. Seligman, et al. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one*, 8(9):e73791, 2013.

[53] Y. R. Tausczik and J. W. Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24–54, 2010.

[54] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

[55] V. N. Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.



NTNU
Norwegian University of
Science and Technology