Importance of Unimportant Words for Authorship Identification

Pedro Filipe da Costa Dias Soares

Thesis to obtain the Master of Science Degree in
Informatic Engineering

Supervisor: Andreas Miroslaus Wichert

Examination Committee
Chairperson: Prof. Miguel Nuno Dias Alves Pupo Correia
Supervisor: Prof. Andreas Miroslaus Wichert
Member of the Committee: Prof. João Carlos Serrenho Dias Pereira

June 2019
Acknowledgments

I would like to say a few words to all those that have helped me on this journey.

Firstly, I would like to thank my parents Abel Soares and Patrocinia Soares and to my brother Sérgio Soares for everything they regularly do for me and for making my life a lot easier. Not only did they help me constantly over the years, they gave me opportunity to focus on my work, and provided me with the motivation needed during the complicated moments. Secondly I would like to express my most sincere gratitude to professor Andreas Miroslau Wichert, for his availability, his precious advice, for supervising my work, for giving me great support and always believing in me. He showed great patience and always motivated me during this journey.

António Varela, who taught me the required knowledge for my degree and my future work.

My colleagues, Renato Nunes, Miguel Viola, Francisco Bom, Luís Rodrigues and Fernando Macedo, for constantly motivating me, and helping me solve the problems I faced.

Finally, to all my teachers who granted me the bases to complete this dissertation.
Resumo

Existe um constante crescimento da quantidade de informação disponível. Esta provém muitas vezes de locais conhecidos aos quais instintivamente confiamos. Porem, alguma da informação com que nos deparamos no nosso dia-a-dia provêm de fontes desconhecidas ou duvidosas. Isto leva à necessidade de averiguação do autor dos textos, por ordem a determinar se a informação é fidedigna.

Contudo, apesar de existirem muitas metodologias para a identificação do autor em textos escritos em Inglês, não houve uma quantidade significativa de estudos feitos para textos escritos na Língua Portuguesa.

Isto leva a que apesar de existirem muitos métodos devidamente testados sobre como determinar o autor de um texto com grande precisão, quando estes métodos são utilizados em textos portugueses, estes geram resultados menos satisfatórios. Para além disso, existem features que são tendencialmente ignoradas por não fornecerem informação relevante sobre o autor nas línguas mais testadas, e que potencialmente poderão ser uma mais valia em textos de outras línguas.

Devido ao facto de não haver muitos estudos para a determinação de autores em textos portugueses, decidimos utilizar textos portugueses como a base do nosso trabalho. Nestes textos iremos não só testar algumas das features mais utilizadas para a determinação do autor em textos escritos em outras línguas, mas também iremos testar a nossa proposta de feature, designada “Uninportant Words”. Esta feature é normalmente ignorada pois não fornece informação relevante sobre o autor. Contudo, visto que estamos a trabalhar com textos escritos em língua portuguesa, esta feature poderá eventualmente gerar bons resultados aquando da sua utilização na tarefa de identificação da autoria de textos.

Palavras-chave: Uninportant Words, Textos Portugueses, Identificação de Autores
Abstract

There is an abundance of documents online and frequently this documents contain information that can be relevant for different applications. However one of the problems associated with online documents is that frequently those documents are anonymous. Although identity cues are scarce in cyberspace, individuals often leave behind textual identity traces. Each author writes in a different way, thus by extracting the features from the text it is possible to identify the author of anonymous texts. It can also be used to determine if a text was written by the person claiming to have written it, or even to try and find the author of a given anonymous text.

To correctly identify an author it is important not only to be able to correctly extract features from texts, but also to determine what are the features most suitable for the identification of the author. For our approach, we will focus on the pattern of distribution of unimportant words, since we believe that each author has a specific distribution that will distinguish himself from any other.

**Keywords:** Uninportant Words, Portuguese Texts, Author Identification
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgments</td>
<td>v</td>
</tr>
<tr>
<td>Resumo</td>
<td>vii</td>
</tr>
<tr>
<td>Abstract</td>
<td>ix</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xv</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xvii</td>
</tr>
<tr>
<td>Nomenclature</td>
<td>1</td>
</tr>
<tr>
<td>Glossary</td>
<td>1</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Objectives</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Thesis Outline</td>
<td>2</td>
</tr>
<tr>
<td><strong>2 Fundamental Concepts</strong></td>
<td>3</td>
</tr>
<tr>
<td>2.1 Human-based vs Machine Learning</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Data Sets</td>
<td>3</td>
</tr>
<tr>
<td>2.3 Types of Text</td>
<td>4</td>
</tr>
<tr>
<td>2.4 Tokenization</td>
<td>4</td>
</tr>
<tr>
<td>2.5 Stylometry</td>
<td>5</td>
</tr>
<tr>
<td>2.5.1 N-grams</td>
<td>5</td>
</tr>
<tr>
<td>2.5.2 Token-level Measures</td>
<td>5</td>
</tr>
<tr>
<td>2.5.3 Syntactic Annotation</td>
<td>5</td>
</tr>
<tr>
<td>2.5.4 Vocabulary Richness</td>
<td>5</td>
</tr>
<tr>
<td>2.5.5 Common Word Frequencies</td>
<td>6</td>
</tr>
<tr>
<td>2.5.6 TF-IDF</td>
<td>6</td>
</tr>
<tr>
<td>2.6 Information Extraction Systems</td>
<td>6</td>
</tr>
<tr>
<td>2.6.1 Finite Automaton</td>
<td>6</td>
</tr>
<tr>
<td>2.6.2 Pattern Matching Systems</td>
<td>6</td>
</tr>
<tr>
<td>2.6.3 Principal Component Analysis</td>
<td>7</td>
</tr>
<tr>
<td><strong>2.7 Text Classification Systems</strong></td>
<td>7</td>
</tr>
<tr>
<td>2.7.1 Sequence Rules with Validation</td>
<td>7</td>
</tr>
</tbody>
</table>
List of Tables

3.1 Results obtained by Ahmed et al. on different test beds .......................... 12
3.2 Feature set combination of Richmond Hong Rui et al. .............................. 16
3.3 Features list of Marcia Fissette ................................................................. 17
List of Figures

3.1 Rules to determine the dominant N-gram ............................................. 15
4.1 System Overview ......................................................................................... 22
4.2 System Details ............................................................................................ 24

5.1 Experimental Results of Individual Features ............................................. 32
5.2 Experimental Results with Individual N-Grams ........................................ 35
5.3 Experimental Results with Combinations of Two N-Grams ..................... 35
5.4 Experimental Results with Combinations of Three N-Grams .................. 37
5.5 Experimental Results with Combinations of Four and Five N-Grams ....... 38
5.6 Experimental Results with one Unimportant Word feature ...................... 39
5.7 Experimental Results with Combination of Two and Three Unimportant Words Features 40
5.8 Experimental Results with Individual Document Content Features ........ 42
5.9 Experimental Results with Combination of Two and Three Document Content Features 43
5.10 Experimental Results with Individual Type of Document Feature .......... 44
5.11 Experimental Results with Combination of Two Type of Document Features 45
5.12 Experimental Results with Combination of Two and Three Type of Document Features 46
5.13 Experimental Results with Individual Type of Narrator Feature ............ 47
5.14 Experimental Results with Combination of Two Type of Narrator Features 48
5.15 Experimental Results with Combination of Three and Four Type of Narrator Features 49
5.16 Experimental Results with Individual Groups of Features .................... 51
5.17 Experimental Results with the combination of Two Groups of Features ...... 52
5.18 Experimental Results with combinations of three Groups of Features ....... 53
5.19 Experimental Results with four and five Groups of Features ................. 54
5.20 Comparison Between the Best Combination of Groups .......................... 55
Chapter 1

Introduction

The amount of information available online increases everyday, and with it the amount of documents with anonymous authors. Information Extraction (IE) and Authorship Analysis (AA) are well established research areas, but there still does not exist any perfect method to discover the authorship of a particular document. The difference between Author Identification and Author Verification is that in Author Identification the main task is of determining the author of a piece of text whereas in authorship verification the objective is of determining if a given author (for whom we have a corpus of writing samples) is also the author of a given anonymous text.

The objective of this work is to address the problem of extracting meaningful features of anonymous texts and determining the most likely author, based on pre-analyzed texts containing the author in question. Normally when extracting such features, unimportant words are removed since they do not give any significant information about the author. These words are words like “the”, “a” and “an”. In our case, we will consider these words since although they may provide very little information regarding the author, they might be enough to determine the author by analyzing the distribution of the semantics of the text. For this identification we assume that each author has a certain pattern in which he uses unimportant words.

Several approaches were proposed in the field of IE and AA, below we will discuss some of them, and will describe their advantages and disadvantages.

1.1 Motivation

Since the appearance of the Internet, a large number of documents has become available to the public. This information can come from trusted sources, but frequently this does not happen. Many of these documents will not have an author, and sometimes the authorship of the document will be claimed by different people. This can cause several problems, and much controversy. As such, with our approach we thrive to achieve a better way to identify or confirm the authorship of a document in order to be able to use it.
1.2 Objectives

One of the major challenges of the document authorship identification problem is the extraction of the most appropriate features for representing the style of an author. Several techniques have been proposed, including attempts to quantify vocabulary richness, function word frequencies and part-of-speech frequencies [1]. This is because humans are creatures of habit, and as such have certain personal traits which tend to persist.

All humans have unique (or near unique) patterns of behavior. We therefore conjecture that certain characteristics pertaining to language, composition and writing, such as particular syntactic and structural layout traits, patterns of vocabulary usage, unusual language usage, stylistic and sub-stylistic features will remain relatively constant [1].

However, there still does not exist a consensus on the best possible set of features to determine a document's authorship. Unimportant words features are often disregarded, since it is considered to not provide any relevant information pertaining to the authorship of the text. However, we do believe that this feature has been misjudged, and that it may prove its usefulness when identifying the author of a text. As such, with our work we plan on determining the viability of the use of unimportant words as a feature in order to determine the authorship of a document. There has been a great number of studies published about the task of authorship identification of English documents. However, in comparison, the number of studies done for Portuguese documents can be considered marginal. As such, as the basis of our work, we will use Portuguese texts. With this approach we plan to determine the viability of not only unimportant words, but also other widely used features that provide very good results in English texts.

1.3 Thesis Outline

This thesis is organized in the following way: Chapter 2 presents the fundamental concepts in order to clearly understand the work that is being presented. Chapter 3 presents the previously done work in the area of author identification, giving us a insight on the current techniques and results of works with similar characteristics. In chapter 4 we talk about our proposed approach, as well as the reasons that led us to pick each component used. Chapter 5 provides us with a collection of experiments and the respective results. Lastly, in chapter 6 is dedicated to their conclusions of our work, as well as suggestions of improvements for future works.
Chapter 2

Fundamental Concepts

2.1 Human-based vs Machine Learning

When building an extraction system there are two common main approaches namely machine learning and human-based.

The human based approach requires deep knowledge of the context that it is being worked on, for the creation of a set of rules for information extraction. This approach requires a great amount of work and time, since the manual construction of this rules is an iterative process, where the rules are written, tested, and changed based on the results. This process will be repeated the necessary times, until the success rate of the extraction is acceptable.

For the machine learning approach the only requirements are an engineer with some knowledge of how to code the extraction rules, and the capacity to tag the training corpus. This tagging consists in determining in each text what information should be extracted by the system. The way the information is extracted will depend on the system and technique used for the extraction, and as such, different systems will provide different results.

2.2 Data Sets

Data sets are extremely important on the task of information extraction. They are required for the process of training and validating the algorithm used. Training data is used in name disambiguity approaches based in supervised machine learning [2] or in the parameter estimation phase [3] to test reference data. As such, it is important to pick an appropriate data set for the task at hand [4].
2.3 Types of Text

There are countless number of possible formats for information to be displayed, however it is important to try, and define the type of text that is being analyzed. As such, we will consider that a text can have three possible formats. Those formats are:

- Structured
- Non-Structured
- Semi-Structured

A text is considered structured when it follows a predefined format, or when the elements follow a predefined order, thus allowing for information to be easily extracted with a set of rules.

A text is considered Non-Structured when it has no structure, as such, it does not obey to any specific format, and the order of its elements is random. Due to these characteristics Non-Structured texts are harder to extract information from, since we can not extract information based on their formation, and the information extraction rules need to be based on information that involves the syntax and semantic relations between words.

Semi-Structured texts are the common ground between structured and non structured texts. These do not posses a rigid structure, thus allowing amongst others for the absence of fields and variable field order.

2.4 Tokenization

Tokenization is the method of splitting text into smaller and meaningful elements. These meaningful elements are called amongst others tokens, phrases and words. The extracted group of tokens act as input for processes like parsing and text mining. It is a part of lexical analysis, and in languages that use inter-word spaces, this approach is fairly straightforward. Tokenization is particularly difficult for languages such as Chinese which have no word boundaries, but it is easy for languages such as English. Usually, the tokenization process occurs at the word level. Yet, to define what is meant by a "word" is sometimes difficult to deal with. Frequently, a tokenizer will use any kind of heuristics, as the ones shown on the example below [5]:

- All adjacent strings of alphabetic characters are always a part of one token. The same is the case for the numbers.

- Tokens may be separated by whitespace characters. These may include punctuation characters, a line break or space.

- The resulting list of tokens may or may not include punctuation and whitespace.
2.5 Stylometry

Stylometry is the development of literary stylistics and assumes that each author has distinctive writing habits. These habits are exhibited in features like vocabulary, sentence complexity and phraseology. Stylometry also assumes that these habits are unconscious and as such, they are difficult to deliberately conceal, and therefore difficult to falsely manufacture. Stylometry pursues the establishment of methods for style feature extraction and associated metrics to assert text similarity[6]. Stylometry uses metrics for features such as fluency, grammar, syntax and spelling. Based on these metrics, it is possible to determine the similarity of anonymous texts and the texts of known authors. As such, to increase the successfulness of determining the authorship based on the stylometry, it is important to have suitable quantities of known-author text available and restricting the number of possible authors. Stylometry techniques also require texts to be comparable in the same environment, such as comparing texts from similar genres, like: poetry, prose, drama etc. There exists more than 1000 proposals for style markers, however, the most reliably successful features have, in general, been function words and n-grams[6].

2.5.1 N-grams

Word n-grams are sequences of n adjacent words. Word n-grams have been used in Natural Language Processing applications, due to the ability to determine the most likely word to follow based on the previous words. This ability to determine the next word is important when it comes to determining the author of a document, since based on previous documents of an author, we can determine the likelihood of that author using certain words, word combination or sentences.

2.5.2 Token-level Measures

Token Level measures consider the sample text as a set of tokens grouped in sentences. Typical measures of this category are word count, sentence count, character per word count, and punctuation marks count[7].

2.5.3 Syntactic Annotation

The use of measures related to syntactic annotation of text provide very useful information for the exploration of characteristics of style. Typical paradigms are normalization count (changing a word that is usually a verb or an adjective to a noun), and counts of the frequency of various syntactic categories[7].

2.5.4 Vocabulary Richness

Various measures have been proposed for capturing the richness or the diversity of the vocabulary of a text and they have been applied mainly to authorship attribution studies. The most typical measure of this category is the type-token ratio $V/N$, where $V$ is the size of the vocabulary of the sample text, and $N$ is the number of tokens of the sample text[7].
2.5.5 Common Word Frequencies

Instead of using vocabulary distribution measures, some researchers have counted the frequency of occurrence of individual words in the sample text. Their calculation is simple, but nontrivial effort is required for the selection of the most appropriate word for a given problem [7].

2.5.6 TF-IDF

TF-IDF stands for Term Frequency-Inverse Document Frequency, and is often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus.

Typically, the TF-IDF weight is composed by two terms:

- The first computes the normalized Term Frequency (TF), which is the number of times a word appears in a document, divided by the total number of words in that document.

- Second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

This measure can be successfully used for stop-words filtering in various subject fields including text summarizing and classification. This is useful since when calculating term frequency, stop words like “a” and “the” are frequent in every document, and therefore would provide no significant information about the author [8].

2.6 Information Extraction Techniques

Although there are a large number of extraction techniques typically they can be grouped in one of the following types.

2.6.1 Finite Automaton

A finite automaton is a model with a set of stages, each stage has a set of transitions and can be either initial stage, final stage or both, and between each stage there is an entry of an alphabet [9].

Finite automaton techniques are widely used in the creation of lexical analyzers for programming languages. There are many information extraction systems that use to some extent finite automatons, such as WIEN[10], SoftMealy[11], and Stalker[12].

2.6.2 Patter Matching Systems

Several information extraction systems have a basis in pattern matching techniques. These patterns can be described with regular expressions and work by checking a sequence of tokens for the presence
of a pattern. WHISK [13] and Rapier [14] are two good examples of these systems.

### 2.6.3 Principal Component Analysis

Principal Component Analysis (PCA) is a dimension-reduction technique that is used to reduce the number of variables in a set while still containing most of the information.

When determining an author a vector is created with every feature. However with a large number of features, the vector may be too large to study and interpret properly. There would be way too many correlations between the features to be considered.

To be able to interpret the data in a meaningful form, it is necessary to reduce the number of variables to an amount easier to interpret. PCA transforms as much of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components. Each first principal component has as much of the variability in the data as possible, and each succeeding component has as much of the remaining variability as possible.

### 2.7 Text Classification Systems

Several works have used classification systems as techniques to extract information [15]. Text classification systems work by splitting the text into tokens (word or sequence of words), and for each token assigning it a class, where each token can have a set of attributes. To use text classification systems to extract information, the system classifies each token with a value corresponding to the confidence it has that that token has the information we are trying to retrieve. When all tokens are classified, the ones with the highest confidence values are the ones picked. This way, it is possible to retrieve information using text classification systems.

#### 2.7.1 Sequence Rules with Validation

Sequence Rules with Validation (SRV) is an information extraction algorithm capable of generating and classifying rules (rules are conditions imposed upon the text, an example would be that the text needs at least 5 adjectives). It is capable of working in Structured, Semi-Structured, and Non-Structured text. In SRV the information extraction problem is viewed as a classification problem where all possible phrases from the text (up to a maximum length) are considered as instances. Every candidate instance in a document is presented to a classifier, and the system assigns to each of these phrases a metric indicating the confidence that the phrase is the correct filler for the information slot we are trying to retrieve [16].
2.8 Features

Stylistic features are the attributes or writing-style markers that are the most efficient in discriminating the authorship. A good amount of stylistic features includes lexical, syntactic, structural, content-specific, and idiosyncratic style markers.

Lexical features are word, or character-based statistical measures of lexical variation. These include style markers such as sentence/line length, vocabulary richness, and word-length distributions.

Syntactic features include function words, punctuation, and part-of-speech tag n-grams.

Structural features, which are especially useful for online text, include attributes relating to text organization and layout. Other structural attributes include technical features such as the use of various file extensions, fonts, sizes, and colors.

Content-specific features are comprised of important keywords and phrases on certain topics such as word n-grams.

Idiosyncratic features include misspellings, grammatical mistakes, and other usage anomalies. Such features are extracted using spelling and grammar checking tools and dictionaries. Idiosyncrasies may also reflect deliberate author choices or cultural differences [17].

2.9 Naive Bayes

Naive Bayes probabilistic classifiers are also commonly used in text categorization since it only requires a small amount of training data to estimate the parameters necessary for classification. The basic idea is to use the joint probabilities of words and categories to estimate the probabilities of categories given a document. Naive Bayes calculates and multiplies the probabilities of all used features to determine the probability of test text. The author with the highest probability is most likely an author of the text. The naive part of such a model is the assumption of word independence. The simplicity of this assumption makes the computation of the Naive Bayes classifier far more efficient than the exponential complexity of non-naive Bayes approaches because it does not use word combinations as predictors [1].

The problem with Naive Bayes is that it can use features that have not been seen in the training data, as such their probability will be zero. To correct this issue a Laplace corrector is often used. This corrector says that the probability of a feature is never zero and as such a value of 1 is added to every count in the data set. This prevents a zero probability situation and ensures that each function has a probability of occurrence based on at least a single count, even if that feature was never been displayed in the training data [18].

2.10 Neural Networks

A neural network is made up of nodes with directed weighted links between them. The network has an input layer representing the input features, an output layer to give the output of the model, and possibly several hidden layers. The weighted sum of the input of a node is used as an input for an activation
function, which determines the output of that node. The activation function makes it possible to produce an output that is a nonlinear function of the inputs. During the learning phase the weights of the network are adjusted until an error rate is minimized. A widely used method to minimize this error is the gradient descent. For training the hidden units a commonly used method is back-propagation. A training example is presented to the network and the network produces an output. This output is compared to the actual class the example belongs to. The weights are adjusted so that the output of the network is closer to the actual output. These methods can only be used in supervised learning and the class labels need to be known in order to learn.

To use a neural network a lot of parameters have to be set: the number of input nodes which depends on the number and type of features, the number of output nodes which depends on the number of classes, number of hidden layers, number of nodes in the hidden layers, the activation function, and the initial weights. Improperly setting these parameters may result in under-fitting so the network can not fully describe the data or in over-fitting so the network can not generalize well to unseen data [19].

2.11 Hidden Markov Models (HMM)

In many cases the patterns that we wish to classify have the tendency to not appear isolated. These sequences of patterns can be used to better classify patterns. When extracting information, we often find cases like this. As such, we can use this for our advantage for better field identification.

Hidden Markov Models [20] are capable of determining the occurrence of patterns in the text, and classifying them in a way that maximizes the probability of a set of patterns, instead of the probability of each isolated pattern.

A Hidden Markov Model is similar to a Markov Model, however the states are hide, and the only thing that can be observed are the symbols emitted by the hidden states. An example of this would be the weather. Normally, we would know if its sunny, cloudy or if its raining just by looking at the sky. In a Hidden Markov Models, these states are hidden, and as such, to determine if its sunny/raining... we need to look at the information emitted by those states. So if it is sunny, the weather is dry, if it is raining, then the roads will be wet, and so on.

2.12 Decision Trees

In decision trees the characteristics of the data are modeled as a tree structure. The root node contains a feature test that separates data samples that has a different value for each feature being tested. Each test should result in subsets of possible categories. The terminal nodes contain the class label. In the case of the task of author identification, this is the name or identification number of the author.

The number of decision trees that can be constructed grows exponentially with the number of attributes. Therefore, an algorithm building decision trees needs to have a strategy to produces a tree within a feasible amount of time. A commonly used strategy is a greedy approach, which creates the nodes of a decision tree by choosing locally the most optimal test. There are several ways to decide what the most
An advantage of decision trees is that once the tree is constructed, classification of unseen data is extremely fast. Another advantage is that when two features are correlated and one is chosen as a test, the other one will not be used anymore. A disadvantage of decision trees is that when the data contains irrelevant features, these might be used in the decision tree, thus creating a tree that is larger than what is required for the classification. This problem can be solved by removing the irrelevant features [19].

### 2.13 Support Vector Machines

Support Vector Machines are based on the structural risk minimization principle from computational learning theory. The idea of structural risk minimization is to find a hypothesis that guarantees the lowest true error for a classification problem. The true error of the hypothesis is the probability that the learned classifier model will make an error on an unseen and randomly selected test example. The values of the features selected for a classification task are transformed into a hyperspace and the support vector machine finds a hyperplane that separates the positive and negative examples from the training set with a maximum margin. The hyperplane that separates the examples is based on a kernel function that can be linear, polynomial, a radial basis function or any other function that the user chooses. The training examples that are closest to the hyperplane and thus define the hyperplane are called Support Vectors [6].

### 2.14 K-fold Cross Validation

K-fold Cross Validation is a way to measure the classifier that is being used. In this method, the data set is divided into k parts. Each of these k parts will be in turn used as test sets and the remaining k-1 parts will be used for training.

For each test set which the classifier correctly classified the predictive accuracy of the classifier for that particular test set is $p = c/k$ [21]. When all test sets have been calculated, the final result will be the average of each of the iterations. This algorithm brings the advantage of more trustworthy results, since its results are not based on a single test that could have been a very fortunate arrangement of training and test sections.
Chapter 3

Related Work

This section presents previous works related with information extraction and author identification. These works addressed similar problems as the one we are facing now. The approaches mentioned below to solve similar problems serve as a cornerstone for our approach, since we can use their results in order to better understand the most viable approach for our system.

3.1 Identification and Similarity Detection in Cyberspace

Faced with the task of identification in cyberspace, Ahmed Abbasi et al.[17] developed the Writeprints technique, which is an unsupervised method that can be used for identification and similarity detection. Writeprints is a Karhunen-Loeve-transforms-based technique that uses a sliding window and pattern disruption to capture feature usage variance at a finer level of granularity.

The technique uses individual author level feature sets where a Writeprint is constructed for each author using the author's key features. For all features that an author uses, Writeprints patterns project usage variance into a lower-dimension space, where each pattern point reflects a single window instance. All key attributes in an author's feature set that the author never uses are treated as pattern disruptors, where the occurrence of these features in an anonymous identity's text decrease the similarity between the anonymous identity and the author.

For the identification task, they compare the Writeprints method against SVM, and for the similarity detection task, they compare it against PCA.

To test this system, information was extracted from four different datasets. The first dataset is composed of email messages from the publicly available Eron email corpus. The second test set consists of buyer/seller feedback comments extracted from eBay (www.ebay.com). The third dataset contains programming code snippets taken from the Sun Java Technology Forum (forum.java.sun.com), while the fourth set of data consists of instant messaging chat logs taken from CyberWatch (www.cyberwatch.com). For each dataset, they randomly extracted 100 authors.

For the identification task, each author's text was split into two identities: one known and one anonymous identity. All techniques were run using tenfold cross-validation by splitting author texts into 10 parts (5 for
known entity, 5 for anonymous identity). The overall accuracy was comprised the average classification accuracy across all 10 folds. The results from his experiments can be seen in table 3.1

With the results obtained in table 3.1 we can conclude that the increased number of authors results in an increasing number of features causing the number of relevant features per authors to decrease, and as such, leading to worse classification results.

### 3.2 Author Extraction Based on Strings of Characters

It is important to notice that when trying to determine the author of a text, every bit of information can be useful. As such Maciej Eder [22] studied how much author information can be retrieved from a raw string of characters in a text sample, without any kind of annotation, parsing, information retrieval, or keyword extracting. For this, the markers he chose were: The most frequent words, word bi-grams, word tri-grams, word tetra-grams, letter bi-grams, letter tetra-grams, letter penta-grams, letter hexa-grams, a combination of words and word bi-grams and a combination of words and letter penta-grams. Then, the retrieved character strings were counted and the obtained numbers were converted to relative frequencies.

To make the results more reliable, a series of parallel attribution experiments were performed on corpora in four languages. The corpora were roughly similar containing seventy prose texts from 20 authors. The languages chosen for his texts were English, Latin, Polish and German.

To test his system he used Burrow’s Delta platform.

By using words and word n-grams as style markers he obtained 100% accuracy for English prose, when having a long vector of words (7500 from the top of frequency list) analyzed. As the number of words in the vector diminished so did the accuracy. When using 6-grams in English prose he obtained around 92% accuracy, and when combining style markers he obtained an accuracy in the order of 95% for English prose.
3.3 Extraction of Author Profile

When analysing a text, there are many available features, and some of those features will have a greater impact in determining some aspect of the text than others. Shlomo Argamon et al. [23] tried to determine what features better help to determine the author’s gender, age, native language and neuroticism level. To do so, he used three different corpora, where age and gender shared the same corpus, each of these corpora where manually labeled according to its category for a particular profiling dimension. For example, when addressing classification by author gender, training documents are labeled as either ‘male’ or ‘female’. Each document was then processed to produce a numerical vector, each of whose elements represents some feature of the text that might help discriminate the relevant categories. A machine learning method then computes a classifier that, to the extent possible, classifies the training examples correctly. Finally, the predictive power of the classifier was tested on out-of-training data.

The corpus for gender and age consisted of the full set of postings of 19,320 blog authors written in English. The (self-reported) age and gender of each author is known and for each age interval the corpus includes an equal number of male and female authors. The texts range in length from several hundreds to tens of thousands of words, with a mean length of 7250 words per author. The tests on this corpora identified the most useful features for gender discrimination to be determiners and prepositions for males, and pronouns for females. Also to note that words related to technology are more used by males, and words related to life or relationships are more used by females. His system achieved an accuracy of 76.1% when determining gender, 77.7% precision for determining the age of the author and also determined that younger writers tend to avoid the use of apostrophes, while that older writers use more determiners and propositions.

The corpus for native language was the International Corpus of Learner English (ICLE) , which was assembled for the precise purpose of studying the English writing of non-native english speakers from a variety of countries. All the writers in the corpus are university students (mostly in their third or fourth year) studying English as a second language. All are roughly the same age (in their twenties) and are assigned to the same proficiency level in English. All texts in the corpus are between 579 and 846 words long. With his system he obtained 79.3% accuracy and noticed that the native speakers of slavic languages (Russian, Bulgarian, Czech) tend to omit the definite article “the” which does not exist in these languages. Features that measure stylistic idiosyncrasies and errors are particularly useful. For example, romanian language speakers often use the vowel ‘o’ where standard English specifies another vowel (e.g., author for author). Regarding content words it is to notice that speakers of different native languages use certain words more than others.

To determine personality, they used essays written by psychology undergraduates at the university of Texas at Austin as part of their course requirements. Students were instructed to write a short “stream of consciousness” essay where they tracked their thoughts and feelings over a 20-minute free-writing period. The essays range in length from 251 to 1951 words. To illustrate personality profiling, they considered just the dimension of neuroticism (roughly: tendency to worry). Here the author obtained an

---

accuracy of 63.1% and noticed that the most discriminating style features indicate that neurotics tend to refer to themselves, use pronouns for subjects rather than as objects in a clause and reflexive pronouns, and consider explicitly who benefits from some action (through prepositional phrases involving, e.g., “for” and “in order to”); non-neurotics, on the other hand, tend to be less concrete and to use less precise specification of objects or events (determiners and adjectives such as “a” or “little”) and show more concern with how things are or should be done (via prepositions such as “by” or “with” and modals such as “ought to” or “should”).

3.4 The Importance of N-Grams to Determine the Author

Vlado Keselj et al. [24] tested a theory of creating n-gram author level profiles, in order to determine the author of a particular text. For his experiment, his dataset consisted of two books for each author, in a total of three authors. With this experiment, he obtained 100% accuracy when using unigrams and a profile size of the 20 more used n-grams. However, when the profile size increased the accuracy dropped to 50%. According to the information from this study, the best profile sizes are between 500 and 3000 n-grams, and the best n-gram size to create these profiles, are 5, 6 and 7-gram. However, the amount of texts used to test this system are low, making it hard to determine if the results that were obtained are trustworthy, and did not happen by luck.

3.5 Decision Trees to Determine the Author of a Text

Zolboo Damiran et al. [25] used decision tree algorithm to classify mongolian literature authors. To build this decision tree, all features were extracted using machine learning. To validate his system, he used 350 documents, belonging to one of 13 authors. His system was trained in 90% of the documents and tested with the remaining 10%. Based on their tests, they obtained an accuracy of above 90%. However, they concluded that the change of percentage of the training and test set influences the results.

John Houvardas et al. [1] used a data set consisting of a training set containing 2500 texts from 50 authors, with 50 messages per author. As features for classification they used the most frequently occurring character n-grams of variable length (3-grams, 4-grams and 5-grams). His proposed method for variable-length n-gram feature selection was to compare each n-gram with similar n-grams (either longer or shorter) and keep the dominant n-grams. To extract the dominant character n-grams in a corpus they used a modification of the algorithm LocalMaxs. Their algorithm used the following rules:

- $g(C)$ is the glue of n-gram C, that is the power holding its characters together.
- $ant(C)$ is an antecedent of an n-gram C, that is a shorter string having size n-1.
- $succ(C)$ is a successor of C, that is, a longer string of size n+1, i.e., having one extra character either on the left or right side of C.
Then, the dominant n-grams were selected according to the rules that can be seen in figure 3.1. Finally, a Support Vector Machine was trained using the reduced feature set, and obtained an accuracy of 73.08%.

### 3.6 Determining the Author Based on a Conversation

Normally, when looking at the problem of determining an author, people analyze papers, books, magazines... However, Marco Cristani et al. [22] experimented on novel features to determine the author of text messages over Skype. The experiments were performed over a corpus of dyadic chat conversations that involved seventy seven subjects. The average number of available words per subject were six hundred and fifteen, and the author identification performance, measured with the area under the cumulative match characteristic curve was 89.5%.

The novel features they implemented were turn duration, writing speed, number of "return" characters, and mimicry (ratio between number of words in current turn and number of words in previous turn).

### 3.7 TF-IDF as an Information Retrieval Method

Bozkurt et al. employed a method which has achieved very high success rates. The method that they employed was the use of Bayesian classifier with stylometric features and function words [26]. They have explored various methods with lexical and syntactic feature sets, with the Bayesian classifier with Gaussian density yielding the best results after applying Principal Component Analysis (PCA). They tested their methods using published documents on a newspaper’s website from 18 different authors, each had written more than 500 articles. These downloaded documents were then parsed with the help of open source Java library HTML Parser, with the pure text data being saved in standardized XML files. Lexical and stylometric features such as number of words, average word length, vocabulary size and number of punctuations were extracted. From their findings, high success rates could be achieved when there are a lot of data available for analysis. Hence, the success rate will increase with the number of texts and samples. In their particular case, the features they used to determine the author were: number of sentences, number of words, average sentence length, average word length, number of different words, number of periods, number of commas, number of colons, number of semicolons, number of exclamation marks, number of incomplete sentences, and number of question marks. Using these features they obtained a success rate of 74.3%.
3.8 Blog Identification

Each type of text analyzed provides challenges, since depending on the environment that people are writing, they will write differently. Richmond et al. focused on the problem of identifying the author on blogs and e-books [21]. For their approach they used standard documents such as e-books to test the functionality of different algorithms, and then used them on online documents (blog entries and forum posts). Their datasets consisted of e-books, with 25 to 50 chapters, blog entries, each containing between 10 and 300 entries, and forum posts, each containing between 20 and 50 posts. In their approach, first the documents were parsed in their software for feature extraction, where they extracted lexical and syntactic features like word counts, unique word counts and frequencies of function words. Then, they used a Bayesian classifier to classify the extracted features. Finally, they analyzed the results and concluded that using Naive Bayes and 62 text files having a total of 125865 words, they obtained an accuracy of 90.3%. As the number of texts reduced the accuracy also reduced, having obtained only 25% accuracy when using just 4 texts in their dataset.

When using lexical features to determine the authorship of the document, they obtained the results shown in Table 3.2.

<table>
<thead>
<tr>
<th>Feature Set (#Features)</th>
<th>Accuracy</th>
<th>Standard Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Lexical Features (4)</td>
<td>58.06%</td>
<td>6.2%</td>
</tr>
<tr>
<td>All Syntactic Features (13)</td>
<td>88.71%</td>
<td>4%</td>
</tr>
<tr>
<td>All Function Words (9)</td>
<td>79.03%</td>
<td>5.2%</td>
</tr>
<tr>
<td>Common Function Words (5) &amp; Vocabulary / No. of Sentences (2)</td>
<td>72.58%</td>
<td>5.6%</td>
</tr>
<tr>
<td>Common Function Words (9) &amp; Vocabulary / No. of Sentences (2)</td>
<td>82.23%</td>
<td>4.8%</td>
</tr>
<tr>
<td>Rare Function Words (4) &amp; Vocabulary / No. of Sentences (2)</td>
<td>64.52%</td>
<td>6.1%</td>
</tr>
</tbody>
</table>

Table 3.2: Feature set combination of Richmond Hong Rui et al.

From their results they concluded that syntactic features such as common stop-function words and vocabulary are good features for large data sets and lexical features such as number of sentences and punctuation are good for small data sets.

3.9 Information Extraction on Short Texts

There is a big difference when extracting the author from long texts and short texts. Marcia Fissette [19] studied how to identify the author of a text given a short text and a list of possible authors. For her work, she relied on the structure of the text and the words that were used. Most of the features used for author identification are stylometric, especially in literary authorship. The data used for her approach was data extracted from a Dutch message board. From that data, there was a selection of 25 messages per author, and a total of 40 authors. This information was then used with the following conditions: word unigram excluding smileys, word unigram including smileys, word bigrams excluding smileys, word bigrams including smileys, dependency triplets, word unigrams and dependency triplets, word bigrams
and dependency triplets. The number of each feature is shown in Table 3.3.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Number of features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unigrams excluding smileys</td>
<td>4008</td>
</tr>
<tr>
<td>Unigrams including smileys</td>
<td>4053</td>
</tr>
<tr>
<td>Bigrams excluding smileys</td>
<td>12692</td>
</tr>
<tr>
<td>Bigrams including smileys</td>
<td>13454</td>
</tr>
<tr>
<td>Triplets</td>
<td>5362</td>
</tr>
<tr>
<td>Unigrams + Triplets</td>
<td>9370</td>
</tr>
<tr>
<td>Bigrams + Triplets</td>
<td>18054</td>
</tr>
</tbody>
</table>

Table 3.3: Features list of Marcia Fissette

To classify the results, Support Vector Machines where used, obtaining on average a success rate of 15.85% on retrieving the correct author.

### 3.10 Minimum Requirements to Determine the Author of a Text

Text length may vary greatly. In many cases is hard to find enough material to conclude if a person really is the author of a specific text. But exactly how much is the actual requirement? Corney et al.[6] used a Support Vector Machines learning method to discriminate between the authorship classes, and through a series of baseline experiments on non-email data. He concluded that 20 text samples of 100 words are sufficient for successful author identification.

After the completion of his baseline experiments, he conducted tests using a corpus of 253 emails messages from 4 authors. The corpus contained over 23,200 words and the email messages varied in length from 0 to 964 words, with an average length of 92 words. He used between 122 and 211 features, and determined that the best single feature for discrimination of authorship has consistently been the set of function words.

Some tests were performed where the different types of function words, such as adverbs, auxiliary verbs, prepositions and pronouns, were used as separate feature sets. The preposition and pronoun function word sets performed slightly better than the others, but better results were obtained when the full set of function words was used. In the end he obtained 99.6% accuracy when using the full set of function words.

Efstathios Stamatatos[7] also studied the problem of identifying authors when there are extremely limited text samples available to be used for training. For his approach he used the CNG method. The CNG method represents each text sample as a bag of character n-grams taking into account case-sensitive information. One of the big advantages of CNG, and the reason it was chosen, is that as a profile-based method, CNG has the advantage of using one big training file per author. So, there is no need for having (or segmenting a big training text into) multiple text samples per author to be used as training set for a machine learning algorithm, such as Support Vector Machines. For his experiment he used texts from Reuters Corpus Volume 1, and adjusted his corpora in several ways, using always 3-grams and CNG. The best results where obtained when all the texts were divided into ten equal
parts (measured in texts per author), and then, nine out of ten formed the training corpus, while the remaining part formed the test corpus. With this approach he obtained a stable accuracy of around 70%, independently of the number of most frequent 3-grams.

3.11 Determining the Author of a Text Using Long Scale Sets

Most studies we have seen so far, have determined the authors of texts using small pools of authors and texts. This gave a clear advantage to their approaches since as the number of authors increase, the number of unique characteristics of each author decreases. This also means that it is difficult to say if the techniques used where good or bad. Unfortunately, for the task of authorship attribution, the amount of people in this world creating content is huge, creating a large pool of possible authors. As such, it is important to determine the feasibility of identifying the author in large scale sets.

Arvind et al.[27] tackled this problem and for their approach they used a dataset comprising over 2.4 million posts taken from 100,000 blogs. In this dataset they extracted a set of features from each post and used them to train classifiers to recognize the writing style of each of the 100,000 blog authors. Most of the information from their dataset was obtained in ICWSM 2009 Spinn3r Blog Dataset, a large collection of blog posts. To clean their dataset, first they removed all HTML and any other markup or software-related debris they could find, leaving only (apparently) manually entered text. Next, they retained only those blogs with at least 7,500 characters of text across all their posts, or roughly eight paragraphs. Non-English language blogs were removed using the requirement that at least 15% of the words present must be among the top 50 English words. To avoid matching blog posts together based on a signature the author included, they removed any prefix or suffix found to be shared among at least three-fourths of the posts of a blog. Duplicated posts were also removed.

At the end of this process, their database contained a total of 2,443,808 blog posts, and an average of 24 posts per blog, where each post contained an average of 305 words, with a median of 223 words. The ten features that had the greatest information gain when computed were:

- the frequency of ""
- the number of characters
- the frequency of words with only first letter uppercase
- number of words
- frequency of non phrases containing a personal pronoun
- the frequency of full stops
- the frequency of all lowercase words
- the frequency of noun phrase containing a single proper noun
- the frequency of all uppercase words
• the frequency of commas

They experimented with Nearest Neighbor (NN), Naive Bayes (NB) and Support Vector Machines (SVM), to determine what was the best classifier for this scenario. For each trial they randomly selected three posts of one blog and set them aside as the testing data. The classifiers were then used to rank each blog according to its estimated likelihood of producing the test posts. They randomly selected three posts of one blog and set them aside as the testing data. The classifiers were then used to rank each blog according to its estimated likelihood of producing the test posts. They only selected blogs from the Spinn3r dataset as the source of test posts, but used the classifiers to rank all 100,000 blogs. In each trial, they recorded the rank of the correct blog.

With this they concluded that SVM’s accuracy drops off rapidly as the number blogs increases. On the other hand NB and NN with a row norm normalization performed surprisingly well, obtaining the correct author in 8% of the cases.

3.12 Determining the Author of Email Texts

One of the most common ways of communication used is emails. Emails can have anonymous authors, and sometimes it is important to determine who exactly wrote a specific email. O.de Vel [28] used Support Vector Machines learning algorithm to determine the author of emails. He used 56 messages from three native English authors. Each author contributed with emails on three topics (about 12,000 words for each author for all topics). The classification was performed using 170 stylistic features and 21 features describing the structure of the email.

When the email topics where aggregated he concluded that the style markers are the dominant features which contribute to the classification performance. The structural features only contribute a maximum of a few percentage points to the classification performance results (changing from 86.2 to 90.5%).

However, because the experiments were executed with messages from only three authors, it can not be concluded that in all author identification tasks the style markers contribute more compared to the structural features.
Chapter 4

Implementation

4.1 Problem Description

This chapter describes the architecture of the proposed approach for determining the author of texts extracted from a website containing technology related articles written in Portuguese. This approach is different from many others since, not only works with Portuguese texts instead of English texts, but also because it explores the possibility of using Unimportant Words as a feature in its' core. The work reported in this chapter can be seen as an advancement over previous approaches, in the sense that it will focus on a language where classification algorithms are normally not tested upon, and because it tests the viability of a normally disregarded feature.

4.2 System Overview

The figure 4.1 illustrates the ideology of our system, namely the systems' input, processing components and output. As shown, the system receives as input, a set of documents that will be spited in training and testing. First, we extract the features of both Training and test documents received as input. Then, the features extracted from the training documents will be used to train our classifier. Finally, in the evaluation module, the previously trained classifier will received the features from the testing documents, and will return his best guess regarding the author of the evaluated test document.

4.3 System Details

In this section we will present in greater detail each of the different components of our solution that can be seen in image 4.2. These are divided into three main components, Inputs, System Components and Outputs.

The inputs component is composed by a Dataset, the List of Features, the List of Classifiers and the Number of Folds:
• Dataset: Our dataset is composed of a group of 600 texts written in Portuguese. These documents were extracted from the website pplware.sapo.pt and the extracted collection consist of 100 documents from each of the six different authors. The subjects and the length of the documents varies arbitrarily, although the general topic of each of the documents is related with technology. Each of the documents extracted were manually processed in order to remove all images, author names and html tags with the objective of leaving only plain text files.

• List of Features: For our approach, the main feature we focused on, is the distribution of unimportant words. Although this feature is normally removed since it is believed that it provides very low to no information regarding the author, we believe it may be enough to identify him, or at least help in the task of author identification, in combination with other features. Thus, as a measure of performance comparison with other related works, we also use other features mentioned in the related work, detailed at section 4.3.1, and include for instance N-Grams, Sentence Length and Number of Words. The flexibility of the developed system allows the user to select any combination of features providing the user the ability of determining which combination yields better results.

• List of Classifiers: Several different classifiers presented on our related work at section 3 have obtained great results. As such, in order to determine which classifier is best suited for the task of author identification for the features available, we tested our approach with several of them. The classifiers made available on the developed system are Naive-Bayes, Support Vector Machines, Decision-Trees, Neural Networks and Back Propagation.

• Number of Folds: In order to test the system, we used 10-fold cross validation. By using 10-fold cross validation we ensure that the results obtained were not due to a coincidence in choosing the training and text sets, since K-fold cross validation determines the mean of the results obtained in each of the test sets. However, the flexibility of the system, allows the user to select any number of desired folds thought the user interface.

The System Components is composed by the Segmentation Module, the Feature Extraction Module, the Training Module and the Evaluation Module:
• Segmentation Module: When our system receives the dataset, it will segment the information into sentences and tokens, which will in turn be used for extracting the desired features, that will be used by the feature extraction module.

• Feature Extraction Module: In this module the segmented text will be used in order to extract each of the features selected by the user. After the features have been extracted, a text document will be created containing values for each of the desired features, and their respective author. These documents will then be written in a Weka framework readable format.

• Training Module: This module receives a Weka training document written by the Feature Extraction Module. The document will be divided into training and test sets. For this we will use the 10-fold cross validation method mentioned previously. The system will be trained with the training section while, the test section will be used later on, to evaluate the trained classifier.

• Evaluation Module: In this module each classifier will be evaluated 10 times (due to the 10-fold cross validation), using the previously created classifier against the respective test sets. The results of each of these evaluations will then be recorded in order to be presented on the Results Report. The Results that will be presented will be the mean of its 10 iterations.

Lastly the Output of the system is the Results Report.

• Results Report: After the evaluation segment has run, a results report will be shown to the user. This report will contain the results obtained by each classifier upon using the selected features. The results report will be a text file containing the name of the classifier, the success rate and the features used.

In a nutshell, in terms of process the system receives a set of documents which are parsed, and all images, author signatures and html tags are removed. The resulting documents will become our dataset. Next, the Segmentation Module will split each document of the dataset into tokens. Afterwards, the Feature Extraction Module uses the tokens from the dataset, to extract the features selected by the user. Then, the system splits the dataset into 10 parts of the same size being one of the groups a test set, and the remaining will be the training set. The features from the training set will then be used by the Training Module, in order to train the selected classifiers.

Finally, the Evaluation Module uses the features of the test set, to try to guess the authors of the test set. The result is recorded and this process is repeated 9 times, one for each fold. Afterwards the result returned will be the average of all the results obtained for each of the K-folds.

### 4.3.1 Selected Features

In our approach we have considered several possible features. However, based on other previous related work, we concluded that the ones that could best help us in our task were the following:

- N-Grams (One-Gram/Two-Gram/Three-Gram/Four-Gram/Five-Gram)
- Unimportant Words
• Position of the First Unimportant Word
• Position of the Last Unimportant Word
• Percentage of Upper Characters
• Percentage of Special Characters
• Amount of Numbers in a Text
• Average Length of Each Sentence
• Number of Sentences of Each Text
• Number of Words of Each Text
• Average Length of Each Word
• Percentage of Sentences that end with Full Stop
• Percentage of Sentences that end with Question Marks
• Percentage of Sentences that end with Exclamation Marks
• Percentage of Sentences that end with Question Marks
• Percentage of Sentences that end with Etc
N-Grams

Upon examining previous works in this field, we have determined not only that N-Grams present good results with the author determination task, but also that increasing the number of the N-gram to values greater than 5 does not provide any significant results. As such, for our approach we have decided to use N-Grams with the length of 1 to 5.

Unimportant Words

A corner stone of our approach was to determine the reliability of unimportant words as a feature in author identification. This feature measures the percentage of unimportant words that exist in a text. For this work, we have used a relatively small number of words, with the objective of expanding this number if positive results were obtained. The words used were: "a"; "o"; "e"; "na"; "no"; "da"; "de"; "do".

Position of the First Unimportant Word

This feature returns the position in a text where the first unimportant word is found. To determine this position we will count how many words are not an "Uninportant Word" starting at the beginning of the text. Different authors will write in different ways, as such the position of the first unimportant word will vary accordingly.

Position of the Last Unimportant Word

This feature returns the position in a text where the last unimportant word is found. To determine this position we will count how many words are not an "Uninportant Word" starting at the end of the text. This way texts from different sizes written from the same author can have the same count. Thus allowing identification of the author no matter the size of the text written.

Percentage of Upper Characters

For this features we calculate the percentage of upper characters in a text. This feature has been previously used in several works in this area, and as such, we also considered this feature as a valid feature in order to determine the author of a text.

Percentage of Special Characters

Like the previously mentioned feature, this feature has been used in previous studies that have focused on the author identification. Since different authors express themselves differently, the amount of special characters will vary accordingly. As such, calculating the percentage of special characters in a text is a feature that will help to determine the author. Examples of some of special characters are "%"; "!"; "("; ")" ...
Amount of Numbers in a Text

Several authors will write texts containing dates and numbers, while others will avoid them. In doing so, the amount of numbers contained in a text will be of great importance to reveal the author of the text. For this feature we are only considering numbers that appear in a numerical format such as "1", "2", etc.

Average Length of Each Sentence

Some authors prefer to write more numerous sentences instead of longer but fewer. Just like many other previously mentioned features, this characteristic reveals their writing habits, and as such, we also used as one of our features in order to determine the author.

Number of Sentences of Each Text

Some texts are clearly bigger than others. However for texts with the same length it is possible to find a big range of sentence length. Since some authors are keen on having short and concise sentences while others prefer to write bigger ones. Thus, the number of sentences of a text becomes a great feature for the identification of the author.

Number of Words of Each Text

This feature provides us with the amount of words contained in a text, and helps us determine the length preference of the author texts.

Average Length of Each Word

The average length of each word is a feature not often used, however, we decided to use it since it can be a good indicative of the author, since some authors will prefer to use smaller and more numerous words for their text, over words with greater length.

Percentage of Sentences that end on Full Stop

Each author has its’ own style of writing, as such the way the author will end each sentence can vary greatly. With this premise, we took an approach that determined the percentage of sentences that ended with a Full Stop.

Percentage of Sentences that end with Question Marks

On their texts, some authors will be very inquisitive towards the reader. As such, many of their sentences will end with a question marks. In regards to this, we calculated the percentage of sentences ending with a question mark, in order to determine the text authorship.
Percentage of Sentences that end with Exclamation Marks

Just like before, some authors will be very affirmative in their texts, as such they will end their sentences with an exclamation mark. Thus, in order to better determine the author of the text, we calculated the percentage of exclamation marks in a text.

Percentage of Sentences that end with Etc

Some of the authors will leave many open sentences while providing information to the user. The result of this can be the end of sentences with etc or ... As such, this feature can also be used in order to determine the author of the text.

4.3.2 Feature Grouping

Since there is not a consensus regarding the best feature or group of features in the analyzed related works, we decided to include several of them in this work, as a way to compare its’ results with our suggested Non Important Words features.

Ideally, all possible combination of the presented features should be tested in order to determine which combination achieves the best result in the tasks of author identification.

However, since testing all the possible combinations and provide a meaningful analysis is impractical, and since the aim of this work is to determine how our proposed feature compares in relation with the most used features, we decided to group the feature into related themes namely:

- Group 1: N-Grams
- Group 2: Unimportant Words
- Group 3: Document Content
- Group 4: Type of Document
- Group 5: Type of Narrator

The goal is to test and analyze all the possible combinations of features within a group and then to compare with the best combination of features of the remaining groups. Finally, the objective is to test all the possible combinations of the best features of each group and perform the same kind of analysis.

The content of each group of features is as follows:

Group 1: N-Grams

This groups is based of only N-Gram features. In our approach we consider only N-Grams of size 1 to 5, since bigger N-Grams revealed no significant results in previous related works. The computation of this features, starts by receiving the size of the N-Grams that we are evaluating, and the texts of the author that are used for training. Then we will create a table containing every N-Gram and the number of occurrences using the training documents.
After creating an N-Gram table containing all the N-grams from the training document and their respective number of occurrences, we will validate each of the test documents. If a testing document contains an N-gram that has not yet occurred in the training document than that occurrence will count as if it has appeared once, in the testing corpus. This measure prevents assigning zero probability for missing N-grams in the corpus.

**Group 2: Unimportant Words**

This group of features is based on our approach in determining the validity of features related with often disregarded unimportant words. For this group we consider the percentage of occurrences in a text of Unimportant Words, the position of the first Unimportant Word and the position of the last Unimportant Word.

**Group 3: Document Content**

The purpose of this group of features is to determine if the content of text that is being analyzed contains any author signature. For this group we determine the percentage of upper characters, the percentage of special characters and lastly the amount of numbers in a text.

**Group 4: Type of Document**

This group of features seeks to determine the type of document that the author tends to write. If the author tends to write short texts with long sentences, or if he tends to write long texts but with short sentences and so on. As such, for this feature group we evaluate the average length of each sentence, the number of sentences of each text, the number of words of each text and the average length of each word.

**Group 5: Type of Narrator**

This set of features intends to determine the type of the narrator that wrote the text. It tries to determine if the author writes in an assertive way, if he makes questions, and so on. As such, for this group the features that we have considered are: the Percentage of Sentences that end with Full Stop, the Percentage of Sentences that end Exclamation Marks, and the Percentage of Sentences that end with Etc.

### 4.4 Evaluation Experiments

This section presents the experiments that were conducted in order to validate the described approach to the task of author identification using Unimportant Words, in Portuguese technological related articles. First, it will be presented the evaluation methodology as well as the the metrics used with the respective description. Then, each experiment and its motivation will be detailed. Lastly, the results of each experiment and their careful analysis is presented.
4.4.1 Evaluation Methodology

In order to evaluate our approach, we compared our proposed feature, Unimportant Words, against groups of features from previous similar approaches in this field.

Our approach focused in Portuguese written technology based articles. Many studies took the approach of author identification, on rigid texts that came from an environment with very specific rules. However, since our texts come from a technology website, the structure and size of each article varies, allowing for more author specific characteristics to be encountered. Lastly, our approach is focused on the use of Unimportant Words, which is an often overlooked feature, since it is believed that it does not provided any relevancy on the task of author identification.

For our approach a set of 600 texts were used for both training and validation, using a 10 fold cross validation. In order to reduce the burden of testing all feature combinations, and to determine the relevancy of Unimportant Words, we divided our features into feature groups and for each of the groups we determined what were the feature or feature combinations that had the biggest impact in the task of author identification.

To do so, the classifiers used for our approach were Decision Tree, Naive Bayes, Support Vector Machines, Back Propagation and Nearest Neighbors. Each feature and feature combination was run through every classifier, and for each document the classifier returns the name of the author it believes is the correct one. Each answer is classified as correct or incorrect, then an average of correct answers is calculated, and the feature or feature combination that displayed the best average is considered the best for each of the classifiers.

The main objective of this is to determine which "Unimportant Words" features are the most relevant, and if these features would be part in the best combination of groups of features, or if their presence would just led to obtaining worst results.
Chapter 5

Experimental Results

This chapter has the objective of describing the experimental results obtained, as well as their respective analysis with the intent of validate the viability of the proposed approach in the task of author identification.

In section 5.1 each feature is tested individually in order to determine which features provide the best results for each classifier.

Then, in section 5.2 we will expand our baseline experiment by grouping the features into related themes, and test all the possible combinations of features and classifiers for each of those groups.

Finally, in section 5.3 we once again expand our approach, by testing all combinations of the best group of features from each group, determined in section 5.2.

5.1 Baseline Solution

For our approach we considered a set of classifiers and features as described in section 4.3.1. The classifiers used were Naive Bayes, Decision Tree, Support Vector Machines, Back Propagation (MultiLayer Perception) and Nearest Neighbors. As a baseline for our experiments, we determined the results of testing each feature individually for each classifier.

This section has the objective of presenting and discussing the obtained results when using these classification algorithms. In order to better visualize the information, the feature names have been abbreviated in the diagrams, as shown below:

- 1-Gram - 1G
- 2-Gram - 2G
- 3-Gram - 3G
- 4-Gram - 4G
- 5-Gram - 5G
- Unimportant Words - NUW
5.1.1 Validation of Each Feature Individually

As it can be seen from figure 5.1, the feature that showed the best results with the Decision Tree classifier was 4G, having an accuracy of 49.21%, followed by 3G with 41.27%, and PSQM and PSE, both had an accuracy of 39.68%.

Figure 5.1: Experimental Results of Individual Features

- Position of the First Unimportant Word - PFUW
- Position of the Last Unimportant Word - PLUW
- Percentage of Upper Characters - PUC
- Percentage of Special Characters - PSC
- Amount of Numbers in a Text - ANT
- Average Length of each Sentence - ALS
- Number of Sentences of each Text - NST
- Number of Words of each Text - NWT
- Average Length of each Word - ALW
- Percentage of Sentences that end with Full Stop - PSFS
- Percentage of Sentences that end with Question Marks - PSQM
- Percentage of Sentences that end with Exclamation Mark - PSEM
- Percentage of Sentences that end with Etc - PSE
Regarding the Naive Bayes classifier, the features that obtained the best results were 4G with 41.27%, PSE with 31.75% and ALS with 30.16% of accuracy.

Support Vector Machines, obtained the best results with 2G and 4G having an accuracy of 28.57%, followed by 5G with an accuracy of 26.98%.

Regarding the Back Propagation classifier, the features that showed best results were PSE with 46.03% of accuracy, followed by 4G with 41.27%, and finally, 3G with 38.10%.

Nearest Neighbor classifier also displayed that 4G was the best feature, with an accuracy of 41.27%, followed by 5G with an accuracy of 36.51% and by PSQM with an accuracy of 34.92%.

In a nutshell, in average the best results were obtained by the Decision Tree classifier. However, this classifier was not much superior than the others since Back Propagation and Nearest Neighbors performed almost as well. On the other hand, Support Vector Machines classifier under performed when in comparison with the remainder of the classifiers. In general, we can conclude that for almost all classifiers 4G, feature was the best one.

It is worth mentioning that although none of the features regarding unimportant words achieved great results, they all obtained results better than random chance which would be 16.6% of accuracy, and that the best classifier varies with the feature itself.

5.2 Groups of Features

Although individual features give us a good estimate of their usefulness, it is important to test them in groups in order to determine if it is possible to obtain better results, and what combination of features outperform the others. As such, in this section we will be grouping the features into themes, and testing them in the task of author identification. The groups of features considered are: N-Grams, Unimportant Words, Document Content, Type of Document and Type of Narrator. The content of each one of these groups is the following:

- **Group 1:** N-Grams is composed by:
  - 1G - 1-Gram
  - 2G - 2-Gram
  - 3G - 3-Gram
  - 4G - 4-Gram
  - 5G - 5-Gram

- **Group 2:** Unimportant Words, is composed by:
  - NUW - Number of Unimportant Words
  - PFUW - Position of the First Unimportant Word
  - PLUW - Position of the Last Unimportant Word

33
5.2.1 Group 1: N-Grams

N-Gram features have been used quite often in the task of author identification. However, it is not only important to test these features in order to create a baseline for our experiment, but also to determine how well they perform in Portuguese texts. Thus, we decide to group N-gram features to check their effectiveness in Portuguese texts.

Experimental Results with Individual N-Grams

The experimental results of individual N-Grams are resumed in Figure 5.2. From this figure we can conclude that in average the best results were obtained by the Decision Tree classifier followed by Back Propagation. These classifiers performed on pair with the remaining ones, having Support Vector Machines being the one that obtained the worst results. However, all these classifiers obtained results better than random chance.

In general, we can conclude that the feature that performed best was 4G, and the feature that performed worse was 3G. Although there was no great difference between feature performances.
Figure 5.2: Experimental Results with Individual N-Grams

Figure 5.3: Experimental Results with Combinations of Two N-Grams
Experimental Results with Combinations of Two N-Grams

Figure 5.3 shows the experimental results of different combinations of two N-grams. Using Decision Tree, the best feature combination was \{1G; 3G\} with 42.9% of accuracy, while \{2G; 4G\} and \{2G; 5G\} both had 42.86% of accuracy. Meaning that in this case it is hard to discern what combination of features will yield better results.

Regarding Naive Bayes, the best features were \{3G; 4G\} with an accuracy of 38.10% followed by \{2G; 4G\} with 36.51% of accuracy.

Support Vector Machines had 33.3% of accuracy with \{2G; 3G\} and 31.75% with \{1G; 2G\}.

Back Propagation had 47.62% of accuracy with \{1G; 4G\} and 44.44% of accuracy with \{2G; 3G\} and \{2G; 4G\}.

For Nearest Neighbors \{1G; 3G\} \{1G; 4G\} and \{2G; 5G\} all had 34.92% of accuracy.

In a nutshell, in average the best results were obtained by the Decision Tree classifier followed by Back Propagation. These classifiers detached themselves from the rest by obtaining considerably better results than the worst performing classifiers. Naive Bayes was the classifier that performed worse in comparison with the other classifiers. Although the results obtained were better than random chance, however it still under performed compared with the remaining classifiers.

In general, we can conclude that the feature combination that performed best was \{2G; 4G\}. Although there was no great differentiation between feature performances.

Experimental Results with Combinations of Three N-Grams

The results of different combinations of three N-Grams can be seen in Figure 5.4. As shown, for the Decision Tree classifier, the combination of \{1G; 2G; 5G\} and \{2G; 4G; 5G\} yielded the best results with an accuracy of 41.27%, although \{2G; 3G; 4G\} also had similar results, with an accuracy of 39.68%.

For Naive Bayes, the combination of features that yielded the best results were \{2G; 3G; 4G\} with 31.75% of accuracy followed by \{1G; 3G; 4G\} with an accuracy of 30.16%.

For the Support Vector Machines, the best results were obtained by the \{1G; 2G; 3G\} feature combination, the \{2G; 3G; 5G\} feature combination, the \{2G; 4G; 5G\} feature combination, and lastly the \{3G; 4G; 5G\}, having all these features combinations obtained an accuracy of 33.33%.

Back Propagation obtained very similar results with all feature combinations, being the \{1G; 2G; 4G\} the best one with an accuracy of 39.68% followed by \{2G; 3G; 4G\} with an accuracy of 38.10%.

Nearest Neighbors best feature combination was \{1G; 3G; 4G\}, and \{1G; 3G; 5G\}, with an accuracy of 34.92% followed by \{2G; 3G; 4G\} with an accuracy of 33.33%.

In a nutshell, in average the best results were obtained by the Decision Tree classifier followed by Back Propagation. Although these classifiers outperformed the remaining ones, they did not provide significant differences in the task of correctly guessing the authorship of the document.

Naive Bayes was the classifier that performed worse in comparison with the remaining classifiers, although the results obtained were better than random chance.
Overall the best performing feature combination was \{2G; 3G; 4G\}, although there was not any significant difference when in comparison with other feature combinations.

Experimental Results with Combinations of Four and Five N-Grams

Since the number of possible combinations with four and five features are small, those results were merged in Figure 5.5.

As shown in Figure 5.5, using Decision Tree, the combination of features that obtained the best results were \{1G; 2G; 4G; 5G\} with 42.68%, followed by \{1G; 2G; 3G; 4G\}, and \{1G; 2G; 3G; 5G;\} with an accuracy of 39.68%.

With Naive Bayes, the most distinguished feature was \{1G; 3G; 4G; 5G\} with an accuracy of 34.39% followed by \{2G; 3G; 4G; 5G\} with an accuracy of 26.98%.

Support Vector Machines obtained an accuracy of 33.33% with \{2G; 3G; 4G; 5G\}, followed by \{1G; 3G; 4G; 5G\} and \{1G; 2G; 3G; 4G; 5G\} with an accuracy of 31.37%.

Back Propagation had an even distribution with \{1G; 2G; 3G; 4G; 5G\} showing the best results with an accuracy of 44.44% followed by \{1G; 3G; 4G; 5G\} with an accuracy of 39.68%.

Lastly, Nearest Neighbors best feature group was \{1G; 2G; 3G; 4G\} with an accuracy of 33.33% followed by \{1G; 2G; 4G; 5G\} and \{2G; 3G; 4G; 5G\} with an accuracy of 31.75%.

In a nutshell, in average the best results were obtained by the Decision Tree classifier followed by Back Propagation. Although these classifiers outperformed the remaining ones, they did not provide significant differences when correctly guessing the authorship of the document. Naive Bayes was the classifier that performed worse in comparison with the other classifiers. Although, the results obtained
were better than random chance, it still under performed compared with the remaining classifiers. Overall, the best performing feature combination was \{1G; 2G; 3G; 4G\}, although there was no significant difference when in comparison with other combinations of 4 and 5 features.

Conclusions of the Experimental Combination of N-Grams

Overall, we can conclude that the best combination of features obtained for N-Grams were, 4G. However, all features performed similarly, being the worst combination the \{1G; 4G; 5G\}. The best performing classifiers was Decision Tree followed by Back Propagation, and the worst performing classifier was Naive Bayes, although there was not a great variation between the classifiers.

5.2.2 Group 2: Unimportant Words

The main focus of this work is to determine if unimportant words can be relevant on the task of author identification. In previous works, features related with unimportant words were often disregarded. Thus, this selection aims to test this group alone, to see how well it performs. We also wish to see if there is any synergy between features of this group, and what combination will obtain the best results.

Experimental Results with Individual Unimportant Words Feature

The experimental results of individual Unimportant words can be seen in Figure 5.6.
When using only one feature, we can determine that for Decision Tree classifier, the best results are obtained by PLUW with an accuracy of 26.98%, followed by PFUW with 23.81% and NUW with 22.22%.

With Naive Bayes classifier, we can see that PFUW obtained the best result with an accuracy of 22.22% followed by NUW with 19.05% and PLUW with 17.46%.

With Support Vector Machines, PLUW obtained a 14.29% of accuracy, followed by PFUW and NUW with 9.52%.

Back Propagation obtained 23.81% of accuracy for NUW, followed by PLUW with 20.63% and PFUW with 15.87%.

For the Nearest Neighbors classifier, PLUW obtained 26.98% of accuracy, while NUW and PFUW obtained only 24.4%.

In a nutshell, in average the best results were obtained by the Nearest Neighbors classifier followed by the Decision Tree. These classifiers outperformed the remaining ones. However, excluding Support Vector Machines, they did not provide significant differences when correctly guessing the authorship of the document. Although Support Vector Machines has drastically under performed the remaining classifiers for most cases, the results obtained were above random chance. Overall, the best performing feature was PLUW. However, there was no significant difference when in comparison any of the remaining features.

It is worth mentioning that the results in itself were not very good for most cases. In some particular cases, the results obtained were worse than random chance. However, in all classifiers, PLUW obtained results better than random chance, meaning that it can stand alone as a feature for the task of author identification.
When combining 2 or 3 features, the results change, since features that before were not that good, can work very well in combination with others. The results of combining two and three Unimportant Words Features can be seen in Figure 5.7.

The results from the Decision Tree classifier reveal that the combination of features \{PFUW; PLUW\} obtained the best results with an accuracy of 28,57%, followed by \{NUW; PFUW\} with 26,98%. Naive Bayes also has \{PFUW; PLUW\} as the best feature combination, with an accuracy of 17,46% followed by \{NUW; PLUW\} and \{NUW; PLUW; PFUW\}, with an accuracy of 12,7%.

Support Vector Machines also had \{PFUW; PLUW\} combination obtaining the best results, with an accuracy of 20,63%, followed by \{NUW; PFUW; PLUW\} with an accuracy of 19,05%.

For Back Propagation, \{PFUW; PLUW\} obtained an accuracy of 33,33% followed by \{NUW; PLUW; PFUW\}, with an accuracy of 31,75%.

Finally, for Nearest Neighbors classifier \{PFUW; PLUW\} obtained an accuracy of 25,40% followed by \{NUW; PFUW\} with an accuracy of 23,81%.

In a nutshell, in average the best results were obtained by the Back Propagation classifier followed by Decision Tree and Nearest Neighbors classifiers. These classifiers outperformed the remaining ones, providing significant differences when correctly guessing the authorship of the document. While for most cases, the results obtained were above random chance, for Support Vector Machines and Naive Bayes, half of the times the obtained results were below random chance.

Overall, the best performing feature combination was \{PFUW; PLUW\}, although there was no significant difference when in comparison with any of the remaining feature combinations.
Conclusions of the Experimental Combination of Unimportant Words Features

Overall, we can conclude that the best combination of features obtained for Unimportant Words were \{PFUW; PLUW\}. All the features performed similarly and the worst feature combination was \{NUW; PFUW\}. The best performing classifier was Nearest Neighbours and with the exception of Support Vector Machines, all classifiers performed on pair.

5.2.3 Group 3: Document Content

In this section we will analyze a group of features related with Document Content. These features were widely used in related works in the task of author identification. In this section, we will determine the validity of these features in Portuguese texts, as well as what feature combinations work best. The features tested in this group are the percentage of upper characters (PUC), the percentage of special characters (PSC) and lastly, the amount of numbers in a text (ANT).

Experimental Results with Individual Document Content Features

The experimental results of document content features can be seen in Figure 5.8. When using only one feature, we can determine that in average the best results were obtained by the Naive Bayes classifier. However, there was no significant difference between the classifiers. The best performing feature was ANT, but once again, there was no significant distinction in the effectiveness of author identification in comparison with other features.

Experimental Results with Combination of Two and Three Document Content Features

The results of the combination of two and three document content features are resumed in figure 5.9. Regarding the Decision Tree classifier, the features that obtained best results were \{PSC; ANT\} with an accuracy of 33,33%, followed by ANT with an accuracy of 30,16% and by \{PSC; PUC\} with an accuracy of 28,57%.

For the Naive Bayes classifier, the features that displayed the best results were \{PSC; ANT\}, with an accuracy of 34,92%, followed by \{PSC; PUC\} with 33,33% of accuracy.

For Support Vector Machines, the best combination of features was \{PSC; PUC; ANT\} with 30,16% of accuracy, followed by \{PSC; ANT\} with 25,40% and PUC with 20,26%.

Just like for Support Vector Machines, Back Propagation’s best feature was \{PSC; PUC; ANT\} with an accuracy of 46,03%, followed by \{PSC; ANT\} with an accuracy of 38,10% and by \{PSC; PUC\} with an accuracy of 36,51%.

For Nearest Neighbors the best feature was \{PSC; ANT\} with 38,1%, followed by ANT with 31,75%, and \{PSC; PUC\} with 26,98% of accuracy.
In a nutshell, in average the best results were obtained by the Back Propagation classifier followed by the Naive Bayes classifier. These classifiers provided significantly better results than the remaining ones in the task of correctly guessing the authorship of the document. The worst classifier was Support Vector Machines, even though for all instances of the test case, the results obtained were above random chance.

Overall, the best performing feature combination was \( \{ \text{PSC}; \text{ANT} \} \), although there was no significant difference when in comparison with any of the remaining feature combinations.

**Conclusions of the Experimental Combination of Document Content Features**

Overall, for the group of features related with Document Content, the best combination of features was \( \{ \text{PSC}; \text{ANT} \} \). All features performed similarly, however when combining features, all feature combinations had a good increase in accuracy in comparison with their performance alone. The classifier that obtained best results was Back Propagation, and the worst was Support Vector Machines. Support Vector Machines classifier distinguished from the remaining classifiers by its low performance, even though for all the features tested, it still obtained better results than the random chance.

**5.2.4 Group 4: Type of Document**

This group focus on determining the importance of type of document features in the task of author identification. The features selected for this group are:
Figure 5.9: Experimental Results with Combination of Two and Three Document Content Features

- Average Length of each Sentence - ALS
- Number of Sentences of each Text - NST
- Number of Words of each Text - NWT
- Average Length of each Word - ALW

These features will be tested individually in order to create a baseline for our experiment, but also to determine how well they perform in Portuguese texts.

Experimental Results with Individual Type of Document Feature

In Figure 5.10 we can observe a summary of the experimental results with Type of Document features. When testing only one feature with the Decision Tree classifier, the best performing feature was NST with 34.92% of accuracy, followed by NWT with 33.33% of accuracy. For Naive Bayes the best performing feature was NWT with 34.92% of accuracy, followed by NST with 30.16%.

Support Vector Machines had an accuracy for NST of 26.98% followed by ALS with 25.40%. Back Propagation best feature was NST with 36.51% followed by NWT with 34.92%. Finally, Nearest Neighbors best feature was NST with 34.92%, followed by ALS with 31.75%

In a nutshell, in average the best results were obtained by the Decision Tree classifier followed by Nearest Neighbors classifier. These classifiers provided significantly better results than the remaining ones in the task of correctly guessing the authorship of the document. The worst classifier was Support Vector Machines, having in one instance of the test case, obtained a result below random chance.
Overall, the best performing feature was NST, although, excluding ALW this feature did not perform significantly better than the others. The ALN feature performed significantly worse than the other features, having in two instances provided results worse than random chance.

Experimental Results with Combination of Two Type of Document Features

Figure 5.11, shows the experimental results of different combinations of two Type of Document Features. For Decision Tree the best performing feature combination was \{ALS; NST\} with an accuracy of 52.38%, followed by \{NST; NWT\} with 46.63%.

For Naive Bayes the best results were obtained by \{ALS; NST\} with 38.10% followed by \{ALS; NWT\} with 34.92%.

Support Vector Machines had \{ALS; NST\} and \{NST; NWT\} both with 31.75% of accuracy, while \{ALS; NWT\} had 30.16%.

In Back Propagation the best feature combination was once again \{ALS; NST\} with 52.38%, followed by \{ALS; NWT\} with 42.86%.

Finally, for Nearest Neighbors, \{NST; NWT\} was the feature that displayed the best results, with an accuracy of 44.44% followed by \{ALS; NST\} with an accuracy of 42.86%.

In a nutshell, in average the best results were obtained by the Back Propagation classifier followed by the Decision Tree classifier. These classifiers provided significantly better results than the remaining ones in the task of correctly guessing the authorship of the document. The worst classifier was Support Vector Machines, having in one instances of the test case, obtained a result below random chance.

Overall, the best performing feature combination was \{ALS; NST\}, and this feature combination outperformed all others. The worst performing feature combination was \{ALS; ALW\}, although this
feature combination never produced results worse than random chance.

Experimental Results with Combination of Three and Four Type of Document Features

Figure 5.12, shows the experimental results of different combinations of three and four Type of Document Features.

For the Decision Tree classifier \{NST; NWT; ALW\} showed an accuracy of 50.79% and \{ALS; NSTALW\} had 47.62%.

For Naive Bayes \{NST; NWT; ALW\} had an accuracy of 46.03% followed by \{ALS; NST; ALW\} and \{ALS; NST; NWT\} with an accuracy of 34.92%.

For Support Vector Machines \{NST; NWT; ALW\} had an accuracy of 42.86%, followed by \{ALS; NST; NWT; ALW\} with an accuracy of 41.27%.

Back Propagation best feature was \{NST; NWT; ALW\} with an accuracy of 47.62%, followed by \{ALS; NST; ALW\} with an accuracy of 46.03%.

Finally, Nearest Neighbors best feature was \{ALS; NST; ALW\} with an accuracy of 47.62%, followed by \{NST; NWT; ALW\} and \{ALS; NST; NWT; ALW\} with an accuracy of 41.27%.

In a nutshell, in average the best results were obtained by the Back Propagation classifier followed by the Decision Tree classifier. These classifiers provided results on par with the remaining classifiers.

Overall, the best performing feature combination was \{NST; NWT; ALW\}, and this feature combination outperformed all others. The worst performing feature combination was \{ALS; NWT; ALW\}, although this feature combination never produced results worse than random chance.
Conclusion of the Experimental Combination of Two and Three Type of Document Features

Based on the present experiments with Type of Document features, we can conclude that the best feature combination in the task of author identification is \{NST; NWT; ALW\}. This feature did not obtain clearly better results than others, although there were some feature combinations that did obtain clearly worst results when in comparison with this feature. The worst feature on this test was ALW, obtaining in some instances results worse than random chance. The best and worst classifier changed according to the number of features used. While for one feature the best classifier was Decision Tree, for two and three features was Back Propagation. It is also worth mentioning that overall, the best performing feature combinations, were combinations of three features, instead of a bigger combination of features.

5.2.5 Type of Narrator

This group will focus on determining the best features to identify the type of narrator. The features that will be used for this task are the Percentage of Sentences that end with Full Stop (PSFS), Percentage of Sentences that end with Question marks (PSQM), Percentage of Sentences that end with Exclamation mark (PSEM), and the Percentage of Sentences that end with Etc (PSE). Afterwards, the experimental results of the different combinations of features will be presented.
Experimental Results with Individual Type of Narrator Features

The experimental results with Type of Narrator features can be seen in Figure 5.13. With the Decision Tree classifier, PSEM obtained the best results, with an accuracy of 38.10%, followed by PSQM with an accuracy of 33.33%.

For Naive Bayes, PSQM obtained an accuracy of 30.16%, followed by PSEM with 28.57%.

For Support Vector Machines, PSEM obtained an accuracy of 15.87% and PSQM obtained an accuracy of 14.29%.

Back Propagation best feature was PSEM with an accuracy of 36.51%, followed by PSQM with an accuracy of 23.81%.

Finally, Nearest Neighbors' best feature was PSEM with an accuracy of 39.68%, followed by PSQM with an accuracy of 36.51%.

In a nutshell, in average the best results were obtained by the Decision Tree classifier followed by the Back Propagation classifier. These classifiers did not provided significantly better results than the remaining ones in the task of correctly guessing the authorship of the document, with the exception of the Support Vector Machines classifier. The Support Vector Machines classifier under performed in comparison with the remaining classifiers, obtaining less than half the accuracy of Decision Tree classifier. The Support Vector Machines classifier has in two instances obtained results bellow the random chance.

Overall, the best performing feature was PSEM. This feature did not provide significantly better results than the remaining features, with the exception of PSFS, which significantly under performed all other features.
5.2.6 Experimental Results with Combination of Two Type of Narrator Features

In this section we have combined two features in order to determine if better results can be obtained in comparison with individual Narrator Features.

The results of this experiment can be seen in Figure 5.14.

For the Decision Tree classifier, the \{PSFS; PSQM\} and \{PSEM; PSQM\} both had the highest accuracy, with 44.44%.

Naive Bayes best feature combination was \{PSFS; PSQM\} with an accuracy of 42.68% followed by \{PSEM; PSQM\} with an accuracy of 38.10%.

Support Vector Machines classifier results, were a lot more uniform than the results from the other classifiers. The best performing feature combinations, were \{PSFS; PSE\}, \{PSEM; PSQM\}, and \{PSQM; PSE\} all with an accuracy of 28.57%.

Back Propagation best feature combination was \{PSQM; PSE\} with 49.21% of accuracy, followed by \{PSFS; PSQM\} and \{PSEM; PSE\} both with an accuracy of 42.86%.

Finally, for Nearest Neighbors \{PSQM; PSE\} was the best feature combination, with an accuracy of 42.86%, followed by \{PSFS; PSQM\} with an accuracy of 41.27%.

In a nutshell, in average the best results were obtained by the Back Propagation classifier followed by the Decision Tree classifier. These classifiers did not provided significantly better results than the remaining ones in the task of correctly guessing the authorship of the document, with the exception of the Support Vector Machines classifier. The Support Vector Machines classifier under performed in comparison with the remaining classifiers, obtaining in one instances results bellow the random chance.

Overall, the best performing feature combination was \{PSQM; PSE\}, although this feature combination did not provided significantly better results than the remaining features.
Lastly, it is important to test combinations of more features to determine if the results obtained in the task of author identification may improve. Figure 5.15 shows the results of testing all possible combinations of three and four features.

With the Decision Tree classifier the best performing feature combination was \{PSFS; PSQM; PSE\} with an accuracy of 50.79\%, followed by \{PSFS; PSEM; PSE\} and \{PSEM; PSQM; PSE\} with an accuracy of 47.62\%. Naive Bayes best feature combination was \{PSFS; PSEM; PSQM; PSE\}, with 41.27\% followed by \{PSEM; PSQM; PSE\} with an accuracy of 39.68\%.

Support Vector Machines best feature combination were \{PSEM; PSQM; PSE\} with an accuracy of 33.33\% followed by \{PSFS; PSEM; PSQM; PSE\} with an accuracy of 31.75\%.

Back Propagation best feature combination was \{PSFS; PSEM; PSQM; PSE\} with an accuracy of 53.97\%, followed by \{PSFS; PSEM; PSE\} with an accuracy of 49.21\%.

Finally, Nearest Neighbors best feature combination was \{PSFS; PSQM; PSE\} with 49.21\% of accuracy, followed by \{PSEM; PSQM; PSE\} and \{PSFS; PSEM; PSQM; PSE\} with 44.44\% of accuracy.

In a nutshell, in average the best results were obtained by the Decision Tree classifier followed by the Back Propagation classifier. These classifiers did not provided significantly better results than the remaining ones in the task of correctly guessing the authorship of the document, with the exception of the Support Vector Machines classifier. The Support Vector Machines classifier under performed in comparison with the remaining classifiers, although it did not obtained any results bellow the random chance.
Conclusion of the Experimental Combination of Type of Narrator Features

Based on our experiments with type of narrator features, we can conclude that the best feature combination in the task of author identification is \{PSFS; PSEM; PSQM; PSE\}. This feature combination did not obtain clearly better results than others, with the exception of PSFS which was the worst feature on the test, obtaining in some instances results worse than random chance. The best classifier changed according to the number of features used. However, independently of the test, the best classifiers were Decision Tree classifier and Back Propagation classifier, being the Decision Tree classifier more predominant. It is also worth mentioning that overall, the best performing feature combinations was \{PSFS; PSEM; PSQM; PSE\}, although the results did not improve in all cases where extra features were added.

5.3 Combinations of Groups of Features

The performance of each group of features is an important milestone for our work. However, the objective of our work is to determine the usefulness of Unimportant Words features, and to determine if the presence of these features may improve the results of other more used groups of features.

Groups of Features

In the previous section, we presented the evaluation and results of each group of features. In this section the feature combinations of each group that achieved the best results in the task of author identification will be compared.

Afterwards, the features that obtained the best results of each group will be combined in order to determine what combination of groups of features performs well together and achieves the best results in the task of author identification.

The best feature combination of each group of features are:

- Group 1 - 4G
  - 4G - 4-Gram

- Group 2 - \{PFUW; PLUW\}
  - PFUW - Position of First Unimportant Word
  - PLUW - Position of Last Unimportant Word

- Group 3 - \{PSC; ANT\}
  - PSC - Percentage of Special Characters
  - ANT - Amount of Numbers in Text
5.3.1 Validation of Each Feature Group

On Figure 5.16 we can see how each group of features performed at the task of author identification.

As we can observe from Figure 5.16, in average the best results were obtained by the Decision Tree classifier followed by the Nearest Neighbors classifier. These classifiers did not provided significantly better results than the remaining ones in the task of correctly guessing the authorship of the document, with the exception of the Support Vector Machines classifier. The Support Vector Machines classifier under performed in comparison with the remaining classifiers, obtaining results worse than random chance when using the features from Group2 or Group5.
Overall, the best performing feature group was Group1 and Group4. This feature groups did not provide significantly better results than the remaining feature groups, with the exception of Group3, where the results were clearly inferior to the remaining groups.

It is worth mention that although Group2 had some results less desirable, it did obtain very good results with the Decision Tree classifier.

5.3.2 Experimental Results with the Combination of Two Groups of Features

Figure 5.17 shows the results of merging two groups of features.

When using Decision Tree the best performing group of features were \{Group3; Group4\} with an accuracy of 44.44\%, followed by \{Group1; Group3\} with an accuracy of 42.86\%.

For Naive Bayes the best performing combination was also \{Group3; Group4\} with an accuracy of 47.62\%, followed by \{Group1; Group2\} with an accuracy of 38.10\%.

Support Vector Machines best performing group feature was also \{Group3; Group4\} with an accuracy of 47.62\%, followed by \{Group1; Group3\} with an accuracy of 34.92\%.

Back Propagation had closer results, but \{Group3; Group4\} was still the best feature group, with an accuracy of 46.03\%, followed by \{Group1; Group5\} with an accuracy of 42.86\%.

Finally, Nearest Neighbors best feature groups were \{Group1; Group2\} and \{Group3; Group4\} both with the same accuracy of 36.51\%.

In a nutshell, in average the best results were obtained by the Decision Tree classifier followed by the Back Propagation classifier. These classifiers did not provided significantly better results than the remaining ones in the task of correctly guessing the authorship of the document.

Overall, the best performing feature group combination was \{Group3; Group4\}. This feature group
combination provided significantly better results than some of the remaining groups. However, some group combinations containing Unimportant words, performed on par with \{Group3; Group4\}.

### 5.3.3 Experimental Results with combinations of Three Groups of Features

Figure 5.18 shows us the results of merging three groups of features.

When using Decision Tree the best group combinations were \{Group2; Group3; Group4\} with an accuracy of 47.62% followed by \{Group1; Group2; Group3\} and \{Group1; Group2; Group5\} with an accuracy of 46.63%.

Naive Bayes best feature group combination was \{Group2; Group3; Group4\} and \{Group3; Group4; Group5\} both with an accuracy of 50.79%.

Support Vector Machines best feature combination was \{Group1; Group2; Group3\} with an accuracy of 53.97%, followed by \{Group1; Group2; Group5\} and \{Group2; Group3; Group4\} with an accuracy of 39.68%.

Back Propagation best performing feature was \{Group2; Group3; Group4\} with an accuracy of 52.38%, followed by \{Group1; Group2; Group3\}, \{Group1; Group3; Group5\} and \{Group3; Group4; Group5\} with an accuracy of 49.21%.

Finally, Nearest Neighbors best performing feature groups were \{Group1; Group2; Group5\} with an accuracy of 44.44%, followed by \{Group1; Group2; Group3\} with an accuracy of 42.86%.

In a nutshell, in average the best results were obtained by the Back Propagation classifier followed by the Decision Tree classifier. These classifiers did not provide significantly better results than the remaining ones in the task of correctly guessing the authorship of the document.
Overall, the best performing feature group combination was \{Group1; Group2; Group3\}. This feature group combination provided significantly better results than some of the remaining groups. This group combination contained the group with Unimportant Words Features.

## 5.3.4 Experimental Results with Four and Five Groups of Features

Figure 5.19 shows the experimental results of merging four and five groups of features.

When using Decision Tree we can see that the best feature group combination was \{Group1; Group2; Group3; Group4; Group5\}, obtaining an accuracy of 52.38% followed by \{Group1; Group2; Group3; Group5\} with an accuracy of 47.62%.

Naive Bayes best feature combination was \{Group2; Group3; Group4; Group5\} with an accuracy of 38.10% followed by \{Group1; Group2; Group3; Group4\} with an accuracy of 34.92%.

Regarding Support Vector Machines, the best performing feature group was \{Group1; Group2; Group3; Group4\} with an accuracy of 42.68% followed by \{Group1; Group2; Group3; Group4; Group5\} with an accuracy of 41.27%.

Back Propagation best feature groups were \{Group1; Group2; Group3; Group4; Group5\} and \{Group1; Group2; Group3; Group5\} both with an accuracy of 50.79%.

Finally, for Nearest Neighbors the best feature combination group was \{Group1; Group2; Group3; Group4; Group5\} with an accuracy of 49.21%, followed by \{Group1; Group2; Group3; Group5\} with an accuracy of 44.44%.

In a nutshell, in average the best results were obtained by the Back Propagation classifier followed by the Decision Tree classifier. These classifiers did not proved significantly better results than the
Figure 5.20: Comparison Between the Best Combination of Groups

remaining ones in the task of correctly guessing the authorship of the document.

Overall, the best performing feature group combination was \{Group1; Group2; Group3; Group4; Group5\}, while the worst performing feature group was \{Group1; Group3; Group4; Group5\}. However, these feature groups did not provide significantly different results as the remaining groups. Although, the results obtained by the best and worst feature group combination were similar, it is important to notice that the only difference was the removal of Group2 features (Unimportant Words Features), and that the presence of this feature group improved the results obtained.

5.3.5 Conclusion of the Experimental Combination of Groups of Features

Figure 5.20 shows the results obtained by the best combinations of one to five feature groups. Based on our experiments with combination of feature groups, we can conclude that the best feature group combination is \{Group1; Group2; Group3\}. This combination contains the following features:

- 4G - 4-Grams
- PFUW - Position of First Unimportant Word
- PLUW - Position of Last Unimportant Word
- PSC - Percentage of Special Characters
- ANT - Amount of Numbers in a Text

This feature group combination did not obtain clearly better results than others, although there were some feature group combinations that did obtain clearly worst results when in comparison with this
combination. The worst feature group combination was Group3, although when in combination with other feature groups, it did prove to be an important feature group.

The best and worst classifier changed according to the number of groups of features used. However, in average the best one is Back Propagation classifier, followed by the Decision Tree classifier. The classifier that obtained worst results was the Support Vector Machines classifier, although there was not a significant difference in the results between the best and the worst classifier.

With this experiment we can conclude that although the Unimportant Words group did not obtain very good results on its own, when in combination with other groups, it provided positive results. We can also conclude that since the Unimportant Words group, is part of the group of features that obtained the best results, this group of features can play an important role in the task of author identification. As such, although alone this group of features seems useless, it should not be disregarded. Thus, making it an important feature to be used in future works, when trying to determine the author of a text.
Chapter 6

Conclusions

In this work, we have thrived to determine the author of Portuguese texts. In order to do so, we have presented a set of new features that are normally disregarded.

The idea behind this set of features, is that each author writes in his own particular and unique way, he is bound to have a unique style of writing “Unimportant Words”. As such, each document will have a unique signature of unimportant words, that can be used in order to better identify the author.

Due to the impracticability of testing all the possible combinations of typically used features plus the ones we proposed, we created groups of features based on their similarity. We evaluated each group in order to determine which features produced better results alone, and which combination of features in each group would provide the best results. Finally, we combined the groups of features to try and understand if the “Unimportant Words” feature group, would have a positive or negative impact on the task of author authentication when combined with other feature groups.

Through our tests we could observe that alone, Unimportant Words features obtained a barely above random chance of identifying correctly the author of a text. However, when this feature was used in combination with others, it increased the chances of correctly identifying the author of a text, and thus proving its usefulness on the task of identifying the author in Portuguese texts.

6.1 Achievements

The main contributions of this work are described in the following subsections.

6.1.1 Creation of a Portuguese Corpus

In order to test our hypothesis, a Portuguese corpus was created based on Portuguese articles, related with technology, published by an online magazine. We started by collecting a set of 600 articles from six different authors. Afterwards, these articles were manually processed in order to remove all images, author names and html tags with the objective of leaving only plain text files.
6.1.2 Proposition of New Features for the Task of Author Identification

In our work, we have proposed three new features to aid in the task of author identification. These features were evaluated individually, and in combination with other features, in order to determine their usefulness.

6.1.3 Strategy to Analyze Features Based in Their Themes

Due to the impracticability of evaluating each possible combination of features, in this work, we have decided to group features based in their similarity. We have evaluated each feature individually, and each possible combination of features in each of their respective groups. Finally, we gathered the best performing features of each group, and evaluated all the possible combinations of groups.

6.1.4 Creation of a Prototype

In our work, we have created a flexible prototype easy to adapt to different features, classifiers and folds.

6.2 Future Work

Despite the good results obtained, there is also a great amount of work to be done.

Firstly, more tests should be done with similar texts, in order to validate the results obtained, and to make sure there results were not due to a coincidence. These performed tests should contain a bigger corpus of texts and authors, in order to confirm the obtained results.

These features should be applied to other fields and languages, in order to determine if the results are not exclusive to the Portuguese language.

All feature combinations should be applied, and not only the best features of each group. It is also necessary to expand the amount of features used when testing.

The unimportant words list should also be expanded to include other non important words, as well as create a similar list for other languages.

Just like Unimportant Words features were disregarded, there may be several other disregarded features, that may achieve positive results when tested in different languages. As such, it is necessary to test these features in each language, in order to determine their importance in the task of author identification.
Bibliography


