

DISSERTATION

Universal Grammar Features for Cross-Language Authorship Attribution

Benjamin Murauer

submitted to the Faculty of Mathematics, Computer
Science and Physics of the University of Innsbruck

in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Advisor: Univ.-Prof. Dr. Günther Specht

Innsbruck, 2022

Abstract

Determining the authorship of a document by analyzing the stylistic choices of authors can be used in digital forensics and is an important field in natural language processing. For authors that write in more than one language, one question that arises is which features of the written texts are being transferred to the respective other language, and if those features can be used to identify the author, independent from the language the features are learned from.

This thesis focuses on this cross-language scenario, and presents contributions in multiple aspects. One big problem in this field is the availability of suitable datasets, since authors writing in multiple languages are relatively scarce. Leveraging internet-scale social media comments, a method is presented that composes datasets using multilingual authors, enabling true cross-language authorship research. In another contribution, a novel type of machine learning feature for cross-language analyses is presented: DT-grams are based on universal grammar features and can be used to effectively classify authors in small-scale attribution problems. Finally, to provide more context to the performance of the DT-grams in other fields, a benchmark for authorship attribution in general is presented, and also experiments in related fields such as authorship profiling are performed.

Zusammenfassung

In der digitalen Forensik als auch in der akademischen Forschung zur natürlichen Sprache ist die Fragestellung der automatischen Feststellung der Urheberschaft von Dokumenten ein relevantes Thema. Dabei werden stilistische Merkmale von Autor:innen analysiert, anhand deren man den Ursprung zuordnen kann. Eine der bisher unbeantworteten Fragen in diesem Forschungsgebiet ist ob mehrsprachige Autor:innen für sie typische Merkmale in mehreren Sprachen verwenden, und ob man solche Merkmale für eine sprachübergreifende Analyse der Urheberschaft verwenden kann.

Diese Arbeit beschäftigt sich mit diesem sprachübergreifenden Szenario, und beinhaltet wissenschaftliche Beiträge in mehreren relevanten Bereichen. Ein großes Problem in diesem Forschungsfeld ist der Mangel an Datensätzen die für diese Art von Forschung verwendet werden können. In dieser Arbeit wird eine Methode präsentiert die durch die Verwendung von Kommentaren aus einer Social Media Plattform Datensätze in verschiedenen Sprachkombinationen zusammenstellt, die für die sprachübergreifende Forschung von Urheberschaft verwendet werden kann. Desweiteren wird ein neues Merkmal vorgestellt welches für das automatische Machine Learning verwendet werden kann: DT-grams verwenden universelle grammatikale Eigenschaften von Texten, die sprachunabhängig berechnet werden können und vor allem in Szenarien mit wenig verfügbaren Daten effiziente Klassifizierungen ermöglichen. Durch die Entwicklung eines ausgiebigen Benchmarks für Urheberschaftsforschung wird ein weiterer Kontext für die Leistung von DT-grams und auch anderen etablierten Methoden geschaffen. Schließlich werden diese Ergebnisse durch Experimente in verwandten Disziplinen wie dem Authorship Profiling ergänzt.

Aknowledgements

Real stupidity beats artificial intelligence every time.

Terry Pratchett

First and foremost, this thesis would not have been possible without the loving support of Julia. I also want to thank my friends and family, who have been patient and understanding over the last years, and helped me get back on track if my motivation was taking a dip, and the regular affirmations of Verena and my parents definitively kept me going. Akita provided me with a boost of oxygen over the last year, and the benefits of the weekly lunches at Granny for my thesis are incontrovertible.

I want to thank my advisor Günther Specht for enabling me to do this project, and the members of the DBIS research group for giving me plenty of discussion material that kicked my thoughts into different tracks. Thank you Eva for providing such a strong scientific base for the entire group, and for always having an open ear. Thank you Mike, for showing me that I should keep an open mind and reminding me that approaches I deemed irrevocable should be questioned and can be changed. And finally, thank you Manfred for your collaborations, contemplations and conversations, which kept luring me into the office.

I also want to thank the Österreichische Akademie der Wissenschaften, who enabled the continuation of my research. Without their support, this thesis would not have been possible.

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt durch meine eigenhändige Unterschrift, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Stellen, die wörtlich oder inhaltlich den angegebenen Quellen entnommen wurden, sind als solche kenntlich gemacht. Die vorliegende Arbeit wurde bisher in gleicher oder ähnlicher Form noch nicht als Magister- /Master- /Diplomarbeit / Dissertation eingereicht.

Datum

Benjamin Murauer

Contents

1. Introduction	1
1.1. Motivation	1
1.2. Research Objectives	2
1.3. Contributions and Publications	4
1.4. Thesis Outline	6
2. Foundations	7
2.1. Authorship Attribution	7
2.2. Features for Text Classification	9
2.2.1. Words	9
2.2.2. Characters	10
2.2.3. n -grams	11
2.2.4. Syntactic Features	12
2.2.5. Lexicographic Frequencies	15
2.3. Feature Representation	15
2.3.1. Bag-Of-Words	16
2.3.2. Embeddings	18
2.4. Classification Models	21
2.4.1. Support Vector Machines	21
2.4.2. Pre-Trained Language Models	24
2.5. Classification Evaluation Metrics	25
2.5.1. Binary Classification Metrics	25
2.5.2. Multiclass Classification Metrics	26
3. Cross-Language Reddit Datasets	29
3.1. Introduction	29
3.2. Related Work	32
3.3. Using Social Media as Data Source	33
3.4. Data Acquisition	34
3.5. Preprocessing	35
3.6. Filtering	36
3.7. Dataset Compilation	37
3.8. Cross-Topic/Genre Datasets	39

3.9. Cross-Language Datasets	40
3.10. Conclusion	43
4. DT-Grams: Dependency-Graph Substructures	45
4.1. Introduction	46
4.2. Related Work	46
4.3. DT-Grams: Dependency Tree Substructures	47
4.4. Language-Independent Grammar Features	51
4.4.1. Summary: DT-grams	55
4.5. Features and Models Using DT-grams	56
4.5.1. Frequencies of DT-grams	56
4.5.2. DT-gram sequences	57
4.5.3. Kernel Methods for DT-Grams	58
4.5.4. DT-gram Embeddings	59
4.6. Finding Optimal DT-gram Parameters	59
4.6.1. Experiment Data	60
4.6.2. Experiment Methods	61
4.6.3. Influence of DT-gram Shapes	62
4.6.4. Influence of DT-gram Dimensions	64
4.6.5. Influence of DT-gram Node Representations	65
4.6.6. Performance of Models and DT-gram Representations	68
4.7. Conclusion	69
5. Evaluating DT-grams on Cross-Language Authorship Attribution	73
5.1. Introduction and Related Work	73
5.1.1. Experiment Setup	75
5.1.2. Strategy 1: Language Independent Features	76
5.1.3. Strategy 2: Multilingual Pre-Trained Language Model	76
5.1.4. Strategy 3: Machine Translation	76
5.1.5. Strategy 4: Combinations	77
5.2. Results	77
5.3. Conclusion	80
6. Evaluating DT-Grams on General Authorship Attribution	81
6.1. Introduction and Related Work	81
6.2. Dataset Characteristics and Metrics	83
6.3. Datasets used in the Benchmark	83
6.3.1. CCAT50	84
6.3.2. CL-Novels	84
6.3.3. CMCC	87
6.3.4. Guardian	89
6.3.5. IMDb62	90
6.3.6. PAN18-Fanfiction	90

6.3.7. Reddit Datasets	91
6.4. Aggregated Scores	92
6.5. Evaluating DT-grams	94
6.5.1. Experiment Setup and Baseline Models	94
6.5.2. Results	95
6.6. Conclusion and Discussion	101
7. Evaluating DT-grams in other Text Classification Fields	105
7.1. Introduction	105
7.2. Authorship Profiling	106
7.3. Conspiracy Detection	108
7.3.1. Dataset and Methodology	111
7.3.2. Results	112
7.4. Conclusion	112
8. Conclusion	115
9. Bibliography	119
Appendices	131
A. DT-grams	133
B. Reddit Comments	134
B.1. JSON structure of Reddit comment	134
B.2. Examples of Excluded Comments	135
C. Authorship Attribution Benchmark Datasets	136
C.1. Project Gutenberg Preamble Example	136
D. Universal Grammar Features	137
D.1. Universal Part-of-Speech Tags	137
D.2. Universal Dependencies	137

Introduction

1.1. Motivation

Authorship attribution denotes the problem of determining the authorship of a document by analyzing its content and comparing it to other documents written by a known set of candidate authors. Besides the academic nature of this type of research, applications in this field range from digital forensics, where the authors of incriminating documents are to be determined, to settling debates involving the uncertainty of the origin of historical documents.

Many different strategies have been developed which tackle this issue from varying standpoints. Among them, studies have shown that including *syntactic information* can help to improve the classification results [94]. In this thesis, the idea of using syntactic features is continued to tackle a special variant of authorship attribution problems: *cross-language authorship attribution (CLAA)*. Here, the known documents from the candidate authors are written in a different language than the document of which the authorship is unknown. Figure 1.1 displays the most important aspect of this task, which is the focus point of this thesis. Summarized to an extreme extent, the objective is to train a model using documents written by several authors, and predict which of these authors has written other documents. The key factor making the problem cross-lingual is that the training and testing documents are written in different languages, by the same authors (i.e., each author writes in multiple languages).

This variant of the traditional authorship attribution problem is of interest in multiple aspects. Firstly, by leveraging documents written in a different language, the amount of training data for a specific problem may be increased. For example, in a forensics application where a potential suspect is known to write in a different language than an incriminating document under analysis, having additional training data for that suspect in that language may improve prediction results. Secondly, CLAA is of fundamental linguistic interest in the

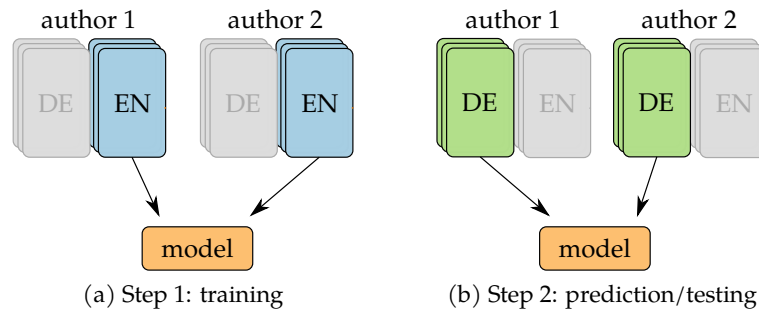


Figure 1.1.: Basic concept of cross-language authorship attribution.

sense that it analyzes which textual features may be typical for an individual, spanning across multiple languages.

From a technical point of view, CLAA imposes profound difficulties on the attribution process, covering many aspects of the overall development. The remainder of this chapter provides an overview of these challenges and explains how this thesis tackles these difficulties in the respective chapters.

1.2. Research Objectives

In order to better understand the research questions that have been driving this thesis, the following paragraph explains the challenges involved in the task of CLAA, and how each of them corresponds to the respective chapter in this thesis. Thereby, details on the general process of authorship attribution and how machine learning is utilized to tackle it are summarized in Chapter 2. The remaining chapters focus on the cross-language aspect of the attribution problem.

At the beginning of most research tasks in natural language processing stands a suitable dataset on which experiments can be performed. In the case of CLAA, acquiring such a dataset is particularly challenging, as it requires authors that write in multiple languages. Previous work in this field relied on translated dataset, where the original authors wrote in a single language and some documents are translated by humans. We identify this as a major resource gap in the field, and the first contribution of this thesis is to close this gap by providing true cross-language datasets that don't rely on human translation, using comments from the social media from internet platform Reddit. Chapter 3 covers this process and explains the details of what previous work exists, why it is not sufficient, and how suitable datasets can be obtained. Sum-

marized, the first requirement of CLAA research is the availability of a suitable *dataset*. This leads to the first research question of this thesis:

RQ1: How can datasets be obtained that are suitable for CLAA?

Having a suitable dataset enables the development of features and models that are capable of cross-language text classification. This thesis uses machine learning to tackle this problem, which includes selecting suitable features to extract from the textual documents, as well as finding classification models that are suitable for the prediction of authorship. Not all approaches that are used in other, single-language text classification tasks can be used in a cross-language setup, so the second requirement for CLAA research is to find suitable *features* and *models*. Here lies the second contribution of this thesis: the development of DT-grams, a cross-language text feature for authorship attribution. It is based on two key concepts: (i) dependency grammar, an approach in which the dependencies of words on one another is modeled, and (ii) universal part-of-speech (POS) tags, a way of representing words using their grammatical role in a language-independent way. The details about the feature development can be found in Chapter 4, which also contains the evaluation of the features on the datasets that are presented in Chapter 3. In summary, DT-grams aim to answer the second research question of this thesis:

RQ2: Which language-independent syntax-based features are a viable choice for a classification feature for CLAA?

While the DT-grams feature was developed specifically for cross-language authorship attribution, the question of how it compares to other text classification features and methods that are used in other related tasks is also relevant in order to estimate the generalizability of the features. Thereby, it is of interest to measure the performance of the novel feature both in single-language authorship attribution setups, as well as in other related text classification tasks. With this challenge in mind, Chapter 6 presents an extensive benchmark involving many different authorship attribution datasets. It aims to focus on different aspects of text datasets such as document length or whether the classification setup is cross-language, and enables the evaluation of features and models in respect to those aspects. This is a novel approach and closes a gap in current research, where novel features and models are often evaluated only on a small number of datasets, and the strengths, weaknesses and the generalizability of those approaches remains unclear. Additionally, two tasks related to authorship attribution, namely authorship profiling and fake news detection, are discussed and experimented with in Chapter 7. In summary, these two chapters of the thesis address the third and final research question:

RQ3: How can approaches in authorship attribution be evaluated in a way that shows their strengths and weaknesses of dataset aspects, and how do the features of RQ2 compare to existing solutions?

1.3. Contributions and Publications

During my time as a Ph.D. student, the following findings have been published in peer-reviewed scientific conferences and workshop proceedings that are related to the topic presented in this thesis:

Conference and Workshop Papers

- Benjamin Murauer, Eva Zangerle, and Günther Specht: A Peer-Based Approach on Analyzing Hacked Twitter Accounts. In *Proceedings of the 50th Hawaii International Conference on System Sciences (HICSS 2017)*, pages 1841-1850. IEEE, 2017. DOI: [10.24251/HICSS.2017.224](https://doi.org/10.24251/HICSS.2017.224)
- Benjamin Murauer, Michael Tschuggnall and Günther Specht: On the Influence of Machine Translation on Language Origin Obfuscation. In *Proceedings of the 18th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2018)*, 2018. DOI: [10.48550/arXiv.2106.12830](https://doi.org/10.48550/arXiv.2106.12830)
- Benjamin Murauer, Michael Tschuggnall and Günther Specht: Dynamic Parameter Search for Cross-Domain Authorship Attribution. In *CEURS Working Notes Proceedings of the 2018 Conference and Labs of the Evaluation Forum (CLEF 2018)*, CEUR No. [2125-84](https://ceur-ws.org/Vol-2125-84/). CEUR-WS.org, 2018.
- Benjamin Murauer and Günther Specht: Generating Cross-Domain Text Classification Corpora from Social Media Comments. In *Proceedings of the 10th International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF 2019)*, pages 114-125. Springer International Publishing, 2019. DOI: [10.1007/978-3-030-28577-7_7](https://doi.org/10.1007/978-3-030-28577-7_7)
- Michael Tschuggnall, Benjamin Murauer and Günther Specht: Reduce & Attribute: Two-Step Authorship Attribution for Large-Scale Problems. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 951-960. Association for Computational Linguistics, 2019. DOI: [10.18653/v1/K19-1089](https://doi.org/10.18653/v1/K19-1089)

- Manfred Moosleitner, Benjamin Murauer and Günther Specht: Detecting Conspiracy Tweets using Support Vector Machines. In *CEURS Working Notes Proceedings of the MediaEval 2020 Workshop*, CEUR No. [2882-10](#). CEUR-WS.org, 2020.
- Manfred Moosleitner and Benjamin Murauer: On the Performance of Different Text Classification Strategies on Conspiracy Classification in Social Media. In *CEURS Working Notes Proceedings of the MediaEval 2021 Workshop*. CEUR-WS.org, 2022 *preliminary proceedings*.
- Benjamin Murauer and Günther Specht: Small-Scale Cross-Language Authorship Attribution on Social Media Comments. In *Proceedings of the 4th Workshop on Technologies for MT of Low Resource Languages (LoResMT 2021)*, pages 11-19. Association for Machine Translation in the Americas, 2021.
- Benjamin Murauer and Günther Specht: Developing a Benchmark for Reducing Data Bias in Authorship Attribution. In *Proceedings of the Second Workshop on Evaluation and Comparison of NLP Systems (Eval4NLP 2021)*, pages 179–188. Association for Computational Linguistics, 2021. DOI: [10.18653/v1/2021.eval4nlp-1.18](#)
- Benjamin Murauer and Günther Specht: DT-grams: Structured Dependency Grammar Stylometry for Cross-Language Authorship Attribution. In *Proceedings of the 32nd GI-Workshop Grundlagen von Datenbanksysteme (GvDB 2021)*, CEUR No. [3075-7](#). CEUR-WS.org, 2022.

Other Contributions

During the course of writing the thesis, I have been part of two collaborations using machine learning to detect the genre of music tracks.

- Benjamin Murauer, Maximilian Mayerl, Michael Tschuggnall, Eva Zangerle, Martin Pichl and Günther Specht: Hierarchical Multilabel Classification and Voting for Genre Classification. In *CEURS Working Notes Proceedings of the MediaEval 2017 Workshop*. CEUR-WS.org, 2017.
- Benjamin Murauer and Günther Specht: Detecting Music Genre Using Extreme Gradient Boosting. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pages 1923-1927. International World Wide Web Conferences Steering Committee, 2018.

1.4. Thesis Outline

The remainder of the thesis is structured as follows: In Chapter 2, the foundations of the technologies that are used in the remainder of the thesis are explained, covering the various textual features and machine learning models included in the later experiments.

Chapter 3 covers the nature of cross-language text classification in detail and describes how existing datasets are not sufficient for cross-language authorship attribution tasks. It further explains how a generic system for constructing such datasets is developed, along with several datasets resulting from that system. This chapter answers the first research question.

Subsequently, DT-grams are introduced in Chapter 4: a family of syntax-based textual features that can be used for inherent cross-language text analysis. This chapter further contains details regarding the parameters that DT-gram offer as well as different strategies on how they can be employed in a machine-learning setup. The chapter contains an evaluation section, which focuses on obtaining optimal parameter values for different datasets.

In order to evaluate DT-grams, Chapter 5 presents different strategies for CLAA in general, and compares the performance of DT-grams to several different approaches.

Chapter 6 provides the means to not only measure the efficiency of the DT-grams features in its intended field of CLAA, but also in other text classification fields by introducing a benchmark for authorship attribution covering a wide variety of different dataset aspects. It also covers the evaluation of the benchmark on different models, including the previously introduced DT-grams.

Afterwards, Chapter 7 puts DT-grams in an even broader context and compares their performance to baseline approaches in several text classification tasks that are related to authorship attribution: fake news detection and authorship profiling.

Related work to each of the mentioned topics is included in the respective chapters.

Finally, the thesis is concluded with Chapter 8.

Foundations

This thesis covers multiple aspects surrounding the task of cross-language authorship attribution. In this chapter, the most important parts of this task are laid out to provide a better understanding of the details in the following chapters. This includes the task of authorship attribution itself, as well as the different steps of the machine learning process that are required to tackle the problem.

2.1. Authorship Attribution

At the heart of this thesis lies the natural language processing task of authorship attribution, which denotes a text classification problem in which the authorship of a document is to be determined. From a set of candidate authors, a classification mechanism has to determine the most likely one. Depending on whether the list of possible answers also includes the option "none of the candidate authors", the attribution problem is called *closed-set* (which guarantees that one of the candidates was the author) or *open-set* (could have been written by a different author that is not in the candidate set). In general, the closed-set variant is easier to solve, and more dominantly researched [43].

Ultimately, the goal of authorship attribution is to find a stylistic property of an author that can be utilized to distinguish the texts from that author from other candidates. This challenge has a long history, and early cases of attribution aimed at finding the true author of controversial manuscripts. A famous example of this is the dispute regarding the authorship of several plays attributed to William Shakespeare [31].

Statistical approaches in this field reach back as far as the late 19th century when Mendenhall [57] analyzed word and letter frequency distributions in novels by different authors. This type of research was applied manually, and

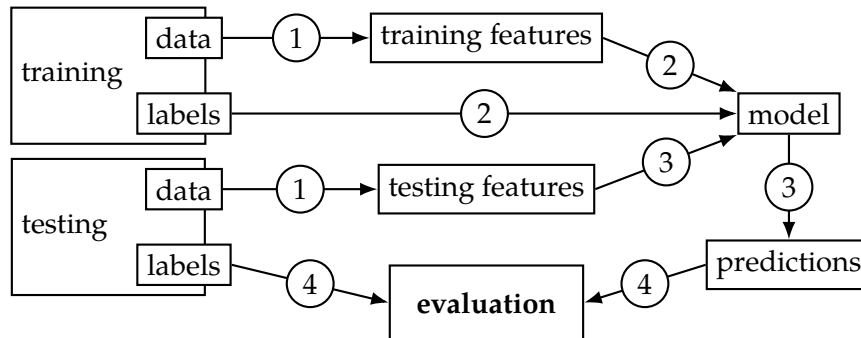


Figure 2.1.: Basic workflow of machine learning models.

therefore could only be performed for a limited number of candidate authors and documents.

Two computer-related developments have transformed the field of natural language processing (NLP) since then. Firstly, the general availability of computers replaces manual calculations, and complex models can be calculated more efficiently. Early examples can be found in the early 1960s, when Mosteller and Wallace [60] use bayesian classification to analyze the Federalist Papers. This is a model frequently found in modern text classification approaches, and it is at this point where the term *machine learning* starts to describe the implemented approaches, although it is difficult to concisely define this concept. Secondly, the availability of large amounts of digital text in the age of the internet has enabled this field of research to generalize using quantitative methods rather than analyzing single instances of authorship debates. Several developments surrounding NLP and text classification specifically rely on vast amounts of data being available for developing and training models. Since then, a wide variety of different textual features and machine learning models have been developed for this task.

In this chapter, the basic concepts behind machine learning with text documents are presented, including foundations on text features, their representations, some classification models, and evaluation methods.

Figure 2.1 shows the basic steps that are performed by a machine learning model. These form the structure of this chapter, as the following parts are explained in each section: Section 2.2 starts with a discussion about which parts of a text can be used as information-carrying features, followed by Section 2.3, which explains how these can be represented numerically and used for machine learning (both sections refer to step 1 in the figure). Thereafter, common classification models and their training process are described in Section 2.4 (step 2). Finally, Section 2.5 goes over how the quality of the predictions of

the model (step 3) can be measured, and how this evaluation is influenced by the nature of the data that is used (step 4).

2.2. Features for Text Classification

When automatically classifying data, measurable numerical features have to be used by the models in question. For some applications, these features are implicitly given by the problem at hand. For example, if a program for optical character recognition should determine which symbol is displayed in an image, the color intensities of each pixel of that image will be used as an input. Other approaches use more abstract features that may be computed from more basic ones. For example, given the same image files as in the previous problem, a calculated feature might be the average curvature of the lines.

However, textual documents usually don't contain intrinsic measurable features like pixel intensities. Instead, one has to choose which numbers are to be calculated from a string of characters that represent a document. In this section, the most important units of information contained in text documents are discussed.

2.2.1. Words

Since written text usually consists of words, one of the most obvious starting points for analyzing written text is to take a closer look at the individual words. Many methods of representing the features in a useful way for the machine learning models (cf. Section 2.3) will just enumerate all words that occur in the set of documents and assign numbers to each, which makes the total number of distinct words in a set of documents relevant for classification performance. This number depends on the language the documents are written in. For example, the Oxford English Dictionary features over 170,000 distinct words for the English language, which are in theory all possible features for a machine learning model¹. Additionally, texts may include further words not included in the dictionary such as proper nouns, misspelled words, or dialect words.

For some models, this large feature space can be a problem both in terms of classification performance as well as runtime, so a desirable preprocessing

¹<http://www.oxforddictionaries.com/words/how-many-words-are-there-in-the-english-language>, accessed on 2022-03-11

word	stem	lemma
swimming	swim	swim
was	wa	be
decentralized	decentr	decentralize

Table 2.1.: Example of stemming and lemmatizing different words.

step is to reduce the number of words that appear in a dataset. One simple solution is to remove all words that don't exceed a specified frequency threshold or occur too frequently. However, this potentially removes useful features. Other approaches try to generalize word forms and bring them to a common base form, and generally can be separated into two groups:

Stemming is a simple rule-based technique that tries to remove pre- and suffixes from words to result in a common base. For example, the words *swimming* would be reduced to *swim*. Some examples of words and their stems can be seen in Table 2.1. Note that these are results for a specific stemming implementation called the *Porter Stemmer* [74], and other stemming algorithms may produce different results.

Lemmatization is a more complex procedure that also considers grammatical variations like different tenses. In contrast to stemming, lemmatization always produces valid words as an outcome, but they might not be valid in a grammatical sense anymore. Examples of different words and their lemmata can be seen in Table 2.1

2.2.2. Characters

Similar to splitting a text into words, one can go a step further and directly look at the atomic parts: its characters. In contrast to the vast amount of possible words for each language, the number of characters usually is far smaller, depending on the language analyzed. The English alphabet contains 26 letters in both upper and lower case, 10 digits, and several punctuation marks. Summed up, the set of possible tokens is orders of magnitudes smaller than for a similar approach using words. The expressiveness of each of these features is now more difficult to explain intrinsically (what can we deduce from a text that contains more "r"s than a different one?), but still a useful feature for many classification tasks.

I_have_been_trying_to_reach_you.

I_h, _ha, hav, ave, ve_, ...

(a) Character 3-grams

I have been trying to reach you .

I-have-been, have-been-trying, been-trying-to, ...

(b) Word 3-grams

Figure 2.2.: Extraction of character and POS tag 3-grams from the same sentence.

2.2.3. n -grams

Single words, tokens, and especially characters are often too short to represent meaningful concepts, and combining groups of multiple instances together can lead to increased performance in many applications. In the context of NLP, these groups are usually referred to as n -grams, where n denotes the length of the sequence and can be applied to words, characters, or any sequence of tokens in a document. Figure 2.2 shows how both character and word 3-grams are extracted from the sentence “I have been trying to reach you”.

The resulting groups of tokens are easy to compute and in practice have been proven to be robust features. For example, character n -grams will capture similar content even if the original text includes spelling mistakes, as only small parts of a misspelled word are affected by a single mistake. Distorting the text by intentionally replacing characters can even help to improve the text classification performance in the case of authorship attribution [87].

By selecting an appropriate value for n , smaller or larger characteristics of the text can be captured with this method. Despite being very simple features, such groups of characters and words have been successfully used in many different text classification applications, including topic detection, authorship classification or sentiment analysis [85, 75]. In these fields, typical values for n are usually between 3 and 5 [42, 86, 73].

2.2.4. Syntactic Features

A different way to look at written text is to take the grammatical structure of the text into account. Therefore, the raw sentences must first be converted to their grammatical representation, which is referred to as *parsing*. This is a wide concept and can range from appending information to each word to detecting sub-phrases or dependencies across different sentences.

POS tags

A typical example of this type of features are POS tags, which represent the grammatical roles of words within a sentence, including *verb*, *noun* or *adjective*. Getting this information from a raw sentence is usually done by using pre-trained models that have been developed specifically for this task. These models are trained on the classification task of *pos-tagging* and are trained on large datasets (called treebanks) containing human-defined ground truth values for all words.

While these roles are generally well-defined within a language, the granularity in which they are used in NLP research differs greatly. For Example, the Brown corpus [45] uses 87 POS tags, whereas the Penn treebank [56] only identifies 36 tags. Thereby, most differences occur in the granularity of the tags (i.e., some punctuations have their own tag in some corpora, but fall into a more general class of “punctuation” in others).

Representing each word with their respective POS tag is a simple method of gaining additional features which can be processed very similarly to words. Such features can be used in conjunction with lexical features, for example by either concatenating them to the feature matrix or by using ensemble methods. The POS tags themselves carry no semantic content, which can be useful for some evaluation applications where avoiding content-based bias, like topic information, is important.

Using POS tags as text classification features is a widely used practice in many NLP fields, including native language detection [10], machine comprehension [49] or authorship attribution [5, 83, 94], to name a few.

Parse Structures

However, replacing the words by different representations doesn't exhaust the full potential of grammar parsing. An additional feature that carries information is the way the words relate to and depend on each other.

These relationships can be expressed by n -grams in a limited fashion, depending on the value of n : when set to a high value, the feature will be able to capture co-occurrences of words and POS tags that appear frequently in a similar distance. However, large values for n drastically increase the feature space, and reasonably small values for n mean that this feature can't capture long-ranging dependencies, where words that occur very late in the sentence refer to words at the beginning.

For example, in the sentence *The vegetables that people often leave uneaten are usually the most nutritious*, the word "are" is referring to the word *vegetables* earlier in the sentence. This relationship can only be grasped by word or POS tag n -grams if $n \geq 7$, which is an uncommonly large value for n which usually is around 3 [86, 42, 90].

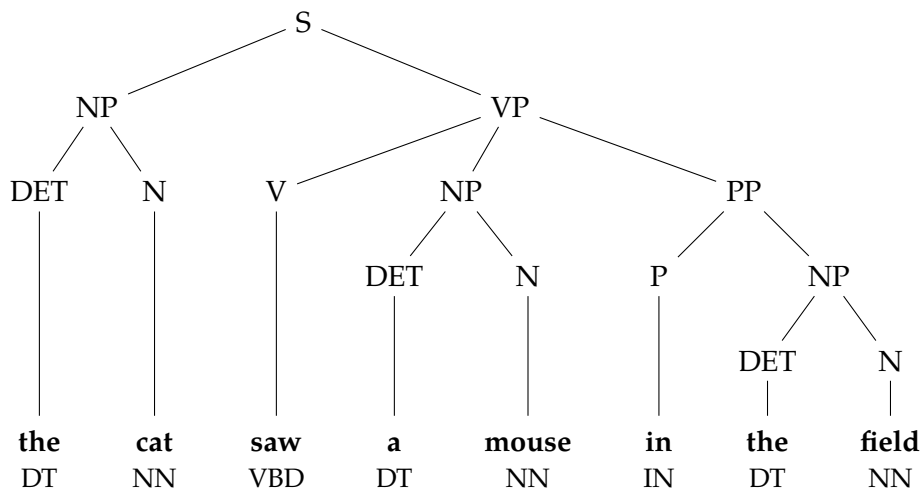
Therefore, alternative means are required to not only capture the neighborhood of words within the sentence but also to consider the proximity of words that are grammatically dependent on one another. One possibility to obtain this information is using grammar parse trees, which represent the grammatical relationships between the words of a sentence in a tree structure. For this purpose, two different representations of sentences have been widely used in previous research:

Constituency parsing breaks down a sentence into sub-phrases or constituents [14]. An example of the constituency parse tree of the sentence *The cat saw a mouse in the field* can be seen in Figure 2.3a.

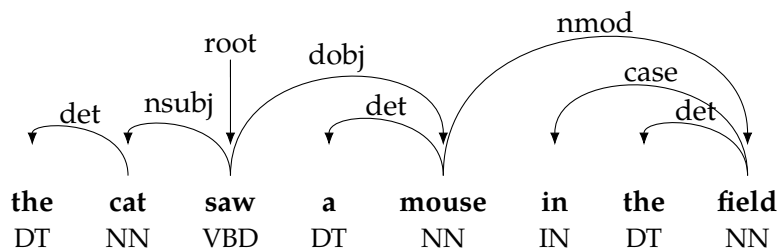
Dependency parsing displays the relationship between different words in a graph structure [29, 65]. For example, in Figure 2.3b, the word *mouse* is dependent on the word *saw*.

Comparing the two representations of the same sentences in Figure 2.3 shows that the dependency graph has many similarities to the constituency tree. Both the constituency tree as well as the dependency graph yield two additional sources of information:

1. They provide means of determining the neighborhood of words. In constituency trees, words that are part of the same constituent can be con-



(a) Constituency parse tree



(b) Dependency graph

Figure 2.3.: Example of constituency and dependency grammar visualizations of the sentence “The cat saw a mouse in the field”.

sidered more related than words in other constituents. Likewise, the number of dependency “hops” between two words in a dependency graph indicates the grammatical distance within the sentence.

2. They provide additional information about groups of words. In the case of constituency trees, sub-phrases of the sentence are labeled with terms like *noun-phrase* or *verb-phrase*, and in dependency graphs, the relationship between two dependent words is labeled.

Upon closer inspection, the two models also show some important differences: In the dependency graph, each relationship between words is labeled rather than the entire substructure that has a specific word as its root. Additionally, the constituency tree contains nodes that don’t have a single corresponding word in the original sentence, while this can’t happen in dependency graphs. For example, in the parses displayed in Figure 2.3, the con-

stituency parse tree contains 14 nodes, while the sentence has only 8 words. The dependency graph, on the other side, will always have as many nodes as there are words in the original sentence.

Further, and more important for this thesis, the dependency grammar does not necessarily care about the order of the words within the sentence, it merely points out the relationships between the words. This becomes vital later in the thesis, where sentences of different languages must be compared, which may use different orders of words and POS tags to express similar concepts.

From these two structures, many different features can be used. For example, the nodes in a constituency graph can be used in addition to the original words to enrich the model with syntactic information (different methods of doing so will be the topic of Section 2.3). Finally, Chapter 4 will make heavy use of the dependency graph in the development of the DT-grams feature.

2.2.5. Lexicographic Frequencies

The ratios and frequencies of specific groups of words in a text have been used as authorship attribution features for many years. For example, the *type-token ratio* calculates the relationship between the number of unique tokens in a document and the number of total tokens that the text contains, therefore representing a measure of vocabulary richness [30]. Another combination is the ratio between function words (pronouns, prepositions, conjunctions, interjections) and content words (nouns, adjectives, verbs, and adverbs), which has been shown to be an expressive feature [25, 2].

2.3. Feature Representation

Having determined which features of a text are used at all is only the first step that is required to prepare a document for a machine learning model. In a further step, a numerical representation of the features must be found that is appropriate for the classification model that is used as part of the machine learning pipeline. There are multiple ways of achieving this, the most important of which are explained in this section.

	and	at	...	zoo
document 1	3	9	...	0
document 2	0	2	...	0
document 3	4	13	...	1

Table 2.2.: Word occurrences in a simple bag-of-words model.

2.3.1. Bag-Of-Words

A simple solution to provide a numerical representation for a text is to count each word in the document, yielding a list of word occurrences. Such an approach is called the *bag-of-words* model and an example is displayed in Table 2.2: for each word in the entire corpus, the number of times the word occurs in each document is counted. The list of occurrences is then used as a number vector representing the document in the remaining machine learning process. This works analogously for characters or n -grams, but for simplicity, the remainder of this section focuses on the case where single words are used.

Its simplicity comes at a cost, however. The most important downsides of the bag-of-words model are:

Sparsity. The amount of possible words for each document to contain is very large, depending on the language of the texts (cf. Section 2.2). Even when considering just the vocabulary that is given by all words that are present in all documents combined, the resulting feature vectors for each document will be very large, and will usually contain many zero values. Such a matrix is called *sparse*, and all the empty values can increase the computation time and negatively impact the classification score for some models.

Neglecting sentence structure. By simply counting word occurrences, the original order of the words and therefore also their relationship to each other is lost, which may contain important information about the document. Depending on the type of classification problem, this may or may not be of importance. For example, a model predicting the mood of a text may be constricted in its expressiveness severely if it would not be able to distinguish the phrases *I'm happy, not sad.* and *I'm sad, not happy.* While the sentences have the same vocabulary, they express exact opposites regarding the author's mood. This special case of dealing with opposites is called *negation detection*, and is difficult to achieve with a simple bag-of-words model.

Importance vs. frequency. The frequency of words within documents usually is not a random or uniform distribution, but follows Zipf's law [55]: while many words occur frequently, across many documents, others are rare and occur only in a few documents. However, this does not necessarily coalesce with the importance of words in the context of text classification. This is especially true for common words that fill grammatical purposes but don't contribute semantically, like articles (e.g., *the*, *a*) or conjunctions (*and*, *or*).

In more detail, Zipf's law suggests that the word *the* is the most common word in the English language, but counting it will be of no use for many applications. On the other hand, words that appear seldom may be of great import may be of great importance. For example, a model classifying the topic of a text may infer the correct topic from a single technical term.

For these reasons, different modifications of the bag-of-words model have been proposed.

Order neglecting can be avoided by no longer looking at single words, but rather counting combinations of multiple words. These combinations are referred to as *n-grams* (or *ngrams*) and can capture more meaning than single words. For example, the sentence *I'm happy, not sad* contains the word 2-grams (or bigrams) *I'm happy* and *not sad*, which are more meaningful than single words when expressing the author's mood. However, by taking combinations into account, the sparsity of the features increases in a combinatoric fashion, depending on the length of the *n-grams*.

Stop Words

Words that are grammatically important, but don't contain any meaning in themselves are called *stop words*. Typical examples of stop words are *the*, *and* or *a*. In many cases, removing stop words from a document can help reduce the total size of the document, as a proportionally large fraction of text consists of stop words. However, for the specific task of authorship attribution, removing stop words has been shown to reduce the performance of prediction models in different languages [70, 77, 27, 80].

TF/IDF

Even without stop words, a document may still be shifted in means of word frequencies. As an example, one application of text classification is to determine an author of a document. If the samples that are available for training the model all have a common topic (e.g., blog posts arguing about politics), many words will appear in a large number of training documents, reducing their expressiveness for the model. Therefore, the tf/idf-model [36] weights each term depending on the occurrence in both each document as well as the number of documents in total:

$$\text{tfidf}(t, d) = \text{tf}(t, d) \cdot \text{idf}(t)$$

$$\text{idf}(t) = \log \frac{|D|}{\text{df}(t)}$$

Here, $\text{tf}(t, d)$ denotes the number of times the term t occurs in the document d , $|D|$ denotes the number of documents in the dataset and $\text{df}(t)$ denotes the number of documents that t occurs in. In practice, the resulting vectors are often normalized using the euclidean norm to remove the bias of the length of documents:

$$v_{\text{norm}} = \frac{v}{\|v\|_2} = \frac{v}{\sqrt{v_1^2 + v_2^2 + \dots + v_n^2}}$$

With this model, terms that occur in many different documents are penalized, whereas terms that appear many times in a few documents are boosted.

2.3.2. Embeddings

As previously described, the bag-of-words model transforms the raw text document into a matrix of word occurrences, which can then be used by different machine learning models. However, some models don't perform well on sparse data, and a more compact representation of words is desired. For example, neural networks generally scale badly in runtime and memory requirements with large sizes of the input vectors. Embeddings are a set of techniques that transform each term of a sequence into a fixed n -dimensional array of numbers, whereas the values of each of these n columns can be interpreted as coordinates in a n -dimensional vector space \mathbb{R}^n . Thereby, any input can be mapped to the fixed-sized vectors, making it a promising tool for reducing the size and sparsity of vectors resulting from word or character n -grams.

word	m -dimensional one-hot vector						word	n -dimensional embedding			
these	0	0	...	0	0	1	these	0.32	0.12	...	0.12
are	0	0	...	0	1	0	are	0.13	0.89	...	0.73
words	0	0	...	1	0	0	words	0.56	0.12	...	0.41
...							...				
in	0	1	...	0	0	0	in	0.72	0.20	...	0.24
document	1	0	...	0	0	0	document	0.87	0.46	...	0.45

(a) One-hot encoded word vectors

(b) Embedded word vectors

Table 2.3.: Difference between one-hot vectors and embedded vector representations of words in a document. Here, m denotes the vocabulary size of the dataset, and n is the (arbitrarily chosen) embedding vector size. Usually, $m \gg n$.

Table 2.3 demonstrates the difference between representing the words in a one-hot vector compared to using embeddings. There are different methods that can be used to obtain the resulting embeddings, and some widely used ones are explained in the following section.

A popular embedding strategy is called Word2Vec and was developed by Mikolov et al. [59], and comes in two variants: CBOW and Skip-gram. In CBOW (depicted in Figure 2.4), a neural network is trained to predict a word w given a set of words that co-occur with w in the original training documents, while in Skip-gram the opposite is the case, where the context of w is to be predicted by the network. Therefore, for each occurrence of w , the words of the context in that occurrence (i.e., the words that surround w) are used for training the network once. Each such occurrence will therefore produce a *training*

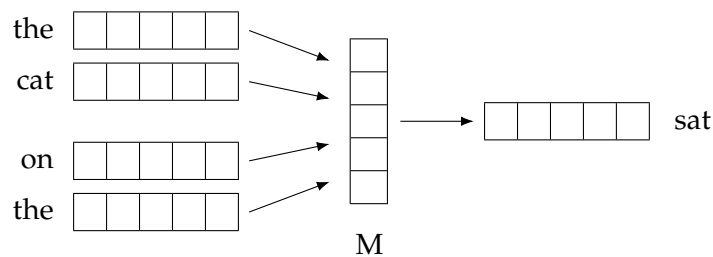


Figure 2.4.: Word2Vec's CBOW embedding model. The context words of the target word *sat* are used to train a network. After training, the embedding coordinates of the target words are in the hidden layer M .

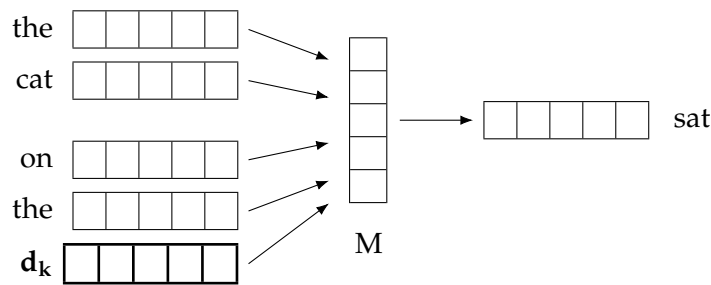


Figure 2.5.: Doc2Vec extension to the Word2Vec CBOW algorithm. The \mathbf{d}_k vector holds the embedded coordinates for the document containing the context used for training.

instance. After training, instead of using the network to actually predict the most likely word surrounding the input, the weights of the nodes in the hidden network layer are used as embedding coordinates for w . This results in words with similar contexts in the training corpus having similar coordinates. Therefore, if the training documents contain many different scenarios a word can appear in, the resulting embeddings will be able to better express this versatility.

This approach has gained popularity among many different NLP applications, partly because the coordinates can be pre-computed efficiently for the widely used purpose of embedding the actual words of the English language by training the network using large amounts of data, yielding high-quality embeddings. However, the approach itself is not dependent on the tokens being words, but rather only requires them to be a set of tokens with a defined context.

Having dense representations of the words within a text is helpful for many NLP tasks, but doesn't directly help machine learning models performing *document classification*, where the stream of words (potentially differing in length for each document) still needs to be transferred to a fixed-sized numeric representation for most classification models. For this purpose, a similar approach to Word2Vec can be used to obtain embedding vectors for entire documents. This approach, called Doc2Vec (or D2V) [46], is a direct extension of Word2Vec. For each training sample of a word w in a document d , a surrogate representation of the entire document is added to the prediction context. Figure 2.5 shows how the CBOW model from Word2Vec is extended by the document matrix: With each training of w , the document vector is added to the context that is used for training the network. All *training instances* that originate from the same document will thereby share the same document vector, which thereby intrinsically learns the words of that document.

A Doc2Vec-like approach is used in this thesis to classify text documents. Further popular embedding approaches focus more on co-occurrences (GloVe [69]) or take the idea of embedding one step deeper and work on a character level (FastText [6]) but were not tested in the scope of this thesis.

2.4. Classification Models

A central part of the machine learning process is the classification model that performs the actual predictions (cf. Figure 2.1). Simplified, it is a mathematical model that learns correlations between a set of training documents and the output class, and can then use the knowledge to predict the output class for previously unseen documents. While the mathematical details of these models are out of the scope of this thesis, the basic principles behind a few selected models are explained in this section, as experiments in this thesis will make use of some of these strategies. This thesis mainly uses two categories of classification models: (1) support vector machines (SVMs) and (2) transformer-based pre-trained language models. The basic functionality of these model families is explained in the remainder of this section.

The experiments in Chapter 4 also include a different type of classifiers called extreme gradient boosting trees (XGBoost) [12]. This classifier was included due to its promising performance in general, measured in various related tasks in preliminary experiments. However, this model is used purely as a black-box classifier to compare the performance of the SVM, and no special properties of the classifier are being exploited in the experiments.

2.4.1. Support Vector Machines

Support vector machines calculate a border between training samples while trying to maximize the distance from each sample to the border. Figure 2.6 displays an example training setting with two different borders, one of which (B) has a larger margin to the samples and is therefore preferred over the other (A). SVMs are binary classifiers, meaning that they can only decide between two output classes. If a classification problem contains more than two classes, a border is calculated for each of the classes independently (one vs. rest).

Let (x_i, y_i) denote the i^{th} training sample of a dataset where x_i contains the features of the sample and y_i denotes the (ground truth) output class. Then the support vector machine is an optimization problem stated by:

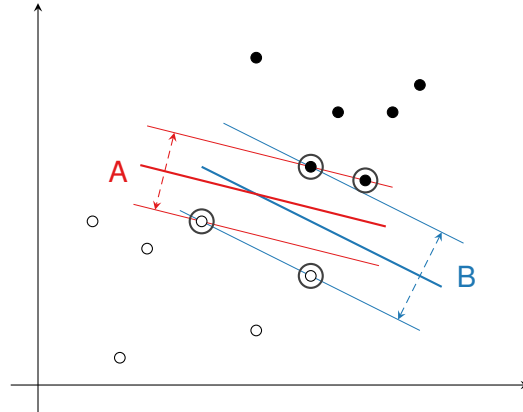


Figure 2.6.: Two different borders between the samples of two classes. Border B has a larger margin (displayed as dashed arrow) between the samples and is therefore the optimal margin for this example.

$$\begin{aligned}
 \max_{\alpha} \quad & W(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m y_i y_j \alpha_i \alpha_j \langle x_i x_j \rangle \\
 \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \dots, m \\
 & \sum_{i=1}^m \alpha_i y_i = 0
 \end{aligned} \tag{2.1}$$

In many cases, some mapping of features ϕ is better able to separate the underlying data patterns than the original features. For example, calculating $\phi(x) = [x^2]$ in addition to the original features x may be required to separate the data sufficiently. In such cases, the kernel trick allows to replace the calculation of the entire feature mapping with a *kernel function* $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$, which replaces the inner product $\langle x_i x_j \rangle$ of Equation 2.1 (or any other classification model that relies on this inner product in its optimization calculation). Thereby, K can often be computed much more efficiently than the entire feature mapping $\phi(x)$. Typical examples of K are:

- $K(x_i, x_j) = x_i^T x_j$ linear kernel
- $K(x_i, x_j) = (x_i^T x_j + c)^d$ polynomial kernel
- $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$ radial basis function kernel

This method can be applied to string documents directly, combining the feature extraction step with the actual classification. Previous research devel-

oped different *string kernels* using this method. In their work called *subsequence kernel* (SKK), Lodhi et al. count common sub-sequences of the documents and use these co-occurrences as entries of the kernel matrix [53]. For a given subsequence length k , the more sub-sequences two documents have in common, the more similar they are, and the higher the corresponding entry in the kernel matrix. Similar approaches using word- instead of character sub-sequences have been proposed by Cancedda et al. [11].

Ionescu et al. have proposed multiple variants of the subsequence kernel for various tasks including native language identification [34, 35] or authorship classification [73]. Thereby, they limit their similarity measurements to *substrings* rather than *subsequences*, meaning that they only consider contiguous parts of the documents, whereas *sub-sequences* can have gaps in them². This restriction has allowed them to implement a very efficient algorithm for calculating the document similarity, with a complexity of $O(\max(|s|, |t|))$ where s and t are the documents being compared.

The following three kernels by these authors are used and adopted as tree-distance kernels, presented later in Chapter 4. In all following equations, Δ^p denotes the set of all sequences of length p using the alphabet Δ . Note that p is a hyper-parameter that can be optimized for a downstream task.

- The spectrum kernel k_p produces a high similarity for documents that have many common sub-sequences in general:

$$k_p(x_i, x_j) = \sum_{v \in \Delta^p} \text{num}_v(x_i) \cdot \text{num}_v(x_j)$$

Here, $\text{num}_v(x)$ denotes the number of times v occurs in x .

- The presence kernel k_p works similar but discards information on *how many copies* of the substrings the respective documents have in common, and only counts how many *distinct* sub-strings co-occur:

$$k_p^{0/1}(x_i, x_j) = \sum_{v \in \Delta^p} \text{in}_v(x_i) \cdot \text{in}_v(x_j)$$

where $\text{in}_v(x)$ is 1 if $v \in x$, and 0, otherwise.

- The intersection kernel k_p^\cap lies between the first two and only considers common sequences important to an extent that occurs in both samples. It is defined as:

$$k_p^\cap(x_i, x_j) = \sum_{v \in \Delta^p} \min(\text{num}_v(x_i), \text{num}_v(x_j))$$

²For example, *pecy* is a *subsequence* of the word dependency as all letters of *pecy* occur in the word in the same order, but it is not a *substring* as it does not occur in the word *without gaps*.

In general, calculating distances using the kernel trick is useful if the kernel method $K(x_i, x_j)$ can be computed faster than the feature mapping $\phi(x)$. Later in this thesis, the kernel trick is used to calculate distances between documents using the similarities of grammatical structures in the documents.

In summary, the kernel trick can be used to calculate features that are more expressive than the linear combination (which is the “default” SVM optimization) without having to calculate a full feature mapping. This gives a performance bonus for scenarios where more expressive features are required. However, this is not always the case, and especially for text classification using sparse features, linear kernels have been shown to not only outperform more complex kernels in respect to runtime but also in terms of classification accuracy [38]. Nevertheless, this thesis includes an approach using the kernel trick in Chapter 4, where it is analyzed whether the features presented in that chapter benefit from the kernel trick.

2.4.2. Pre-Trained Language Models

A fundamentally different approach is taken by *transformer-based language models*. These are a type of neural network that, in the scope of this thesis, is seen as a black-box model. This family of machine learning models has gained much popularity in a wide range of NLP tasks, including text generation [8], dialogue systems [9, 98] but also text classification [89]. Generally, they are used in two stages:

1. Pre-training. In this stage, the model is trained with relatively simple tasks like completing a sentence: Given all but the last word, the model must predict the last word in a sentence. These tasks are performed with vast amounts of data. For example, the GTP3 model [8] was trained using five datasets, which combined contain 500 billion tokens. The goal of the pre-training step is to produce a model that, for a lack of better words, “understands” the language it was trained with by showing the model all possible combinations of words belonging together, in the respective contexts.
2. Fine-tuning. After the pre-training, the model can be used for different tasks by training it a second time, with task-specific samples. Thereby, the second training will not override the previously obtained parameters, but modify them in a way that optimizes the task objective. For example, a pre-trained model can be provided with text classification samples leading the model to combine the previously obtained knowledge to perform the classification tasks.

	predicted positive	predicted negative
actual positive	true positive (TP)	false negative (FN)
actual negative	false positive (FP)	true negative (TN)

Table 2.4.: Binary classification confusion matrix.

In this thesis, several pre-trained models are used in the experiments, with the main purpose of comparing the features and datasets developed in this thesis to a baseline approach.

2.5. Classification Evaluation Metrics

The performance of a machine learning classification model can be measured by comparing the predicted output classes to a previously known ground truth: the closer the predictions match the actual classes, the better the model performs. There are many different ways this comparison can be computed, not all of which are suited for all types of classification problems and datasets that are used for training.

In this section, the terms TP, FP, TN, and FN denote the outcome of a model predicting whether a sample x belongs to the class y , which can be displayed in a confusion matrix (cf. Table 2.4).

Depending on the setup of the experiment, different metrics can be used that use the above definitions in various ways.

2.5.1. Binary Classification Metrics

The widely used *accuracy* metric is calculated as follows:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

It is a simple metric (what fraction of samples are classified correctly?), but it is unsuited for scenarios where the distribution of classes is unbalanced. For example, if 90 samples are of class "A", and 10 of class "B", then a model which blindly predicts every sample to be "A" will obtain an accuracy score of 0.9, but won't be of any practical use.

Therefore, in this thesis, the more restrictive F_1 score is used for most experiments, which represents the harmonic mean between *precision* and *recall*. Thereby, precision denotes the fraction of true positives to all positives (i.e., “How many of the positive predictions by the model are actually true?”), whereas recall denotes how many of the total true samples were predicted as such (i.e., “How many of all positive samples did the model find?”) [96]:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Then, the F_1 score is defined as:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

Note that in general, depending on the classification task, precision and recall should not be weighted equally. For example, misclassifying a healthy person to have a serious illness may have vastly different consequences than declaring an ill person healthy. In these cases, more specific evaluation scores should be used that focus on the requirements of the experiment semantics.

2.5.2. Multiclass Classification Metrics

The previous metrics can be used to calculate the score of a model for predicting a specific output class. When the experiment only has two possible classes, the values for these classes are directly related to each other. For example, a model that has a high F_1 score for detecting spam emails will also have a high F_1 score for detecting valid (non-spam) emails. However, this relationship is no longer valid for experiments where there are more than two possible outcome classes. There are different strategies on how to mitigate this problem and still be able to provide a single measurement that represents the performance of a model on the entire dataset over all classes involved.

Macro-averaging denotes averaging the output of a metric for each class, regardless of how many samples that class includes. It weighs each *class* equally.

Micro-averaging denotes averaging the output of each sample, regardless of what class it originally belonged to. It weighs each *sample* equally.

To illustrate this difference, imagine the output of a classifier for a problem consisting of 100 samples in three output classes *A*, *B* and *C*:

		predicted		
		A	B	C
actual	A	5	0	0
	B	0	2	0
	C	0	70	23

The F_1 -scores of this scenario are calculated as follows:

class	TP	FP	FN	F_1
A	5	0	0	$\frac{2 \cdot 5}{2 \cdot 5 + 0 + 0} = 1.000$
B	2	70	0	$\frac{2 \cdot 2}{2 \cdot 2 + 70 + 0} = 0.054$
C	23	0	70	$\frac{2 \cdot 23}{2 \cdot 23 + 0 + 70} = 0.397$

Then, the micro- and macro-averaged F_1 scores are calculated as follows:

$$F_1(\text{macro}) = \frac{1.0 + 0.054 + 0.397}{3} = 0.483$$

$$F_1(\text{micro}) = \frac{2 \cdot (5 + 2 + 23)}{2 \cdot (5 + 2 + 23) + (0 + 70 + 0) + (0 + 0 + 70)} = 0.300$$

It can be seen that while the accuracy score is the same for both averaging techniques, the micro averaged F_1 score is lowered by the many samples of class *C* compared to the macro-averaged F_1 score, which weighs each class the same, resulting in a significantly higher value.

In general, it is not possible to recommend one version over the other, as different experiments require different evaluation strategies. For example, treating each class regardless of the number of samples (i.e., macro-averaging) may

be desired or even required when the generalizability of a model is evaluated on an unbalanced dataset.

In the remainder of this thesis, the experiments are designed so that in as many cases as possible, the experiments use balanced datasets, where the values for macro- and micro-averaged F_1 scores are identical.

Cross-Language Reddit Datasets³

In the previous chapter, the technical means to measure textual features and classify documents are presented. This chapter contains the first cornerstone of this thesis and describes the development of datasets that are required for evaluating *cross-language* authorship attribution experiments, tackling the first research question of this thesis. After some technical terms are introduced, the chapter continues to explain important differences to existing work and thus motivates the necessity of additional resources. A framework able to compose such datasets from social media data is then described in detail and followed by a presentation of several datasets that have been constructed using that framework, which will be used in evaluation experiments in later chapters.

3.1. Introduction

Having a dataset to perform experiments with is a fundamental resource for any NLP research. Therefore, the first step in cross-language authorship attribution is to find a suitable dataset that fulfills all requirements. However, the field of cross-language text analysis is not well-defined, and many studies (which will be analyzed in detail shortly) interpret the terms which are involved differently. While many definitions of them sound very similar on a quick glimpse, the evaluation and experiment setup used in the respective works are difficult to compare to one another. This is especially true when talking about a classification problem (and therefore, the dataset and strat-

³Results and contents of this chapter are based on and partially reused from my paper: Benjamin Murauer and Günther Specht: *Generating Cross-Domain Text Classification Corpora from Social Media Comments*. In 20th Conference and Labs of the Evaluation Forum (CLEF'2019), pages 114-125, 2019. It has been extended with further explanations, and the names of the datasets have been adopted to match the remainder of the thesis.

egy used for evaluation) being *multi*-lingual, which is a focus of this thesis. In this section, these differences are analyzed, and subsequently, the unsuitability of all existing datasets is explained through detailed descriptions of their respective approaches and differences to the requirements we impose on a suitable dataset.

This thesis contains many descriptions of datasets, which often use terms that are not universally agreed upon. To mitigate misconceptions and confusions regarding some terms when talking about datasets, this thesis will use some fixed terms that are defined as follows:

(i) Dataset terminology used in this thesis	
definition	description
D	a dataset (a set of documents)
d	a document
$\text{auth}(d)$	the author of document d
$\text{lang}(d)$	the language of document d
\mathbf{d}_a	$\{d \mid \text{auth}(d) = a\}$, the set of all documents by author a
A_D	$\{\text{auth}(d) \mid d \in D\}$, the set of all authors of dataset D
\mathbf{l}_a	$\{\text{lang}(d) \mid d \in \mathbf{d}_a\}$, the set of languages used by author a
$\#(d s)_{w c}$	the length of document (d) or sentence (s) measured in words (w) or characters (c)
$ X $	number of items in set X
$\sigma(Y)$	the standard deviation of the values in Y
\bar{Y}	the arithmetic mean of the values in Y

Now we are able to distinguish previous research from this thesis with the help of the following terminology of datasets A dataset is defined as:

mono-language if the entire dataset contains only documents of a single language: $|\{\text{lang}(d) \mid d \in D\}| = 1$

mixed-language if the dataset contains documents in different languages, but not exclusively from authors writing in more than one language (e.g. it contains German documents from author a_1 and English documents from author a_2): $|\{\text{lang}(d) \mid d \in D\}| > 1$.

cross-language if it contains multiple authors that have written documents in more than one language, and can be split in such a way that the

training documents and testing documents have different languages: $|\{\text{lang}(d) \mid d \in D\}| > 1$, and $|\bigcap_{a \in A_D} \mathbf{l}_a| \geq 1$.

strictly cross-language if all authors have written documents in the same different languages: $|\{\text{lang}(d) \mid d \in D\}| > 1$, and $\sigma(\bigcap_{a \in A_D} \mathbf{l}_a) = 0$. This scenario enables evaluations in multiple directions⁴ (e.g., train with language A and test with language B, and vice-versa), whereas the non-strict cross-lingual may be restricted to one evaluation direction.

Examples of these definitions are provided in Table 3.1. In some cases, more information than the dataset alone is required to determine the nature of the evaluation scenario. For example, some datasets consist of sub-datasets which are mono-language datasets each (an example is the PAN18-FF dataset included later in Chapter 6). In these cases, it depends on whether a classification model is re-used for all sub-tasks, or a specific model for each task is used to determine the nature of the evaluation scenario. In the former case, the scenario is considered mixed-language, as the model learns similar features from different languages, but applies this knowledge to documents of the same language. In the latter case, multiple mono-language scenarios contribute to the overall experiment.

As a bare minimum, the CLAA experiments in the remainder of this thesis rely on a *cross-language* dataset. If the dataset is additionally *strictly* cross-language, additional comparisons can be evaluated depending on which documents are used for training and testing, respectively. As will be shown in the next section, existing datasets that are widely used in the field don't meet these requirements, so a novel approach to providing this resource is required. This

⁴In practice, "multiple" is virtually always restricted to 2, as datasets containing multiple authors who all write in the same three languages have not yet been composed due to the lack of data.

dataset type	$\max_{a \in A} \mathbf{l}_a $	example author languages		
		\mathbf{l}_{a_1}	\mathbf{l}_{a_2}	\mathbf{l}_{a_3}
mono-language	1	{en}	{en}	{en}
mixed-language	≥ 1	{en}	{en, de}	{de}
cross-language	> 1	{en, de}	{en, es}	{es, fr}
strictly cross-language	> 1	{en, de}	{en, de}	{en, de}

Table 3.1.: Different types of datasets used for authorship attribution. $|\mathbf{l}_a|$ denotes the number of different languages of all documents written by author a .

task is tackled in the remainder of this chapter and answers the first research question of this thesis: *How can datasets be obtained that are suitable for CLAA?*

3.2. Related Work

With the help of these terms, we can now analyze the most important existing studies in this field:

Stuart et al. [88] use a mixed-lingual dataset where each author writes documents in only one language, either English or Russian. They analyze the effectiveness of simple, low-level features in a mixed authorship attribution setting containing documents in English, Russian, and transliterated Russian, mapping the Cyrillic to the Latin alphabet. In this scenario, the machine learning model does not necessarily need to detect any stylistic features from the authors but can achieve classification results by detecting the language of the document, which restricts the set of candidate authors.

Eder et al. [19] compare the performance of various feature families for different languages. Therefore, they utilize several mono-lingual attribution datasets, where the model always receives training and testing documents of the same language. For each language, a different model (albeit with the same hyperparameters) is trained. Similarly, Kestemont et al. [40] constructed a dataset consisting of multiple sub-problems in different languages.

As the closest match to the requirements of CLAA, Bogdanova et al. [5] and Llorens [52] both use cross-language⁵ human-translated datasets. Thereby, an author writes documents only in one language A. Then, part of these documents are translated by humans into language B, and are then used as test documents for the attribution problem.

Even if the original versions of the test documents (in language A) are not in the training set, the argument holds that the original author of the documents did not write them in different languages. Therefore, any analysis using this data doesn't perform cross-language authorship attribution, but rather measures how well the translation process obfuscated the single-language attribution problem. Additionally, the influence of the human translator becomes a factor that must be considered, and although previous studies suggest that it is negligible [97], the fact remains that the original author only produced text

⁵Bogdanova et al. use a strictly cross-lingual dataset containing Spanish and English texts, while the dataset from Llorens et al. contains authors that only overlap in one language.

in one language, making statements about a language-independent writing style questionable.

When summarizing the existing work, a lack of proper cross-language authorship analysis datasets emerges. This gap is addressed in this chapter, where a strategy is presented to compose true cross-language datasets without relying on human translation.

3.3. Using Social Media as Data Source

The aim of this thesis is to analyze the ability of language-independent text features based on universal grammar concepts to distinguish multilingual authors in authorship attribution. More specifically, we want to determine to which extent grammatical features are kept across different languages when authors write in different languages. For this reason, we reject all previous interpretations of a cross-language attribution setup: The first two don't refer to a classification setup where authors provide documents in multiple languages at all, and the third interpretation uses translated datasets. However, the latter problem, according to Bogdanova et al. [5] is less an active experimental design decision rather than a direct consequence of a dire lack of true cross-language datasets.

We recognize this as a fundamental gap in resources that influences any research in this area. Therefore, we developed a method to create datasets sourced from social media comments [61] that allows to analyze *true* cross-language authorship attribution problems.

The rapidly growing size of internet communication allows for more elaborate data collections to be extracted. In particular, the social media platform Reddit⁶ contains vast amounts of text which are freely available online. From this resource, we have constructed a method that allows us to compose datasets containing texts in multiple languages from many authors. It allowed us to create a dataset by selecting bilingual authors for different language pairs, enabling for the first time true untranslated multilingual authorship analyses. The software and its documentation are available online⁷.

⁶<https://reddit.com/>

⁷https://github.com/bmuraueer/reddit_corpora



Reddit terminology used in this thesis

post: a link, text, or image that a user posts to Reddit. This represents a root content, that does not require any previous related content.

comment: a text that is added to a post as an additional comment. Comments are always bound to one post or to a previous comment.

subreddit: an area of coherent posts. This coherence can be topical (i.e., all posts must cover the same subject), language-based (e.g., all posts must be written in Dutch), serve a specific purpose (e.g., in */r/AskReddit*^a, users can ask arbitrary questions to the Reddit community), ...

^aIn this thesis, the subreddit with the name “AskReddit” (which is located at <https://reddit.com/r/AskReddit>) is referred to as */r/AskReddit* for better distinction from other terms.

While the focus of this thesis lies on cross-language authorship attribution, the presented solution for composing datasets is also able to compose⁸ cross-genre and cross-topic datasets of a single language, as well as mixed languages. In the remainder of this chapter, all steps of the composition process are described in detail.

3.4. Data Acquisition

At the beginning of this composition, process stands a dump of all comments posted on Reddit⁹. It contains over 3 billion comments from more than 22 million authors (see Table 3.2). Each comment has a maximum length of 10,000 characters and is represented as a json object with a plethora of meta information. An example of a comment in this format is given in Appendix B.1. Most notably, each comment was written inside a *subreddit* (or *sub*), which represents a mostly topic-related subspace. However, it must be noted that the concrete moderation of each subreddit differs and that usually even less

⁸The term *compose* is used in the context of this thesis to emphasize that the documents themselves are not generated. Instead, the focus lies on selecting which documents from a vast pool of candidates are chosen to produce a useful result set.

⁹<https://files.pushshift.io/reddit/comments/>, accessed in May 2019

	pre-filter	post-filter
comments	3,092,028,928	50,567,575
authors	22,554,169	4,380,330
subreddits	415,566	162,564

Table 3.2.: Raw Reddit data statistics.

strict requirements are enforced on the content of the comments. For example, discussions about a certain post can often drift to a completely different topic, making the entire dataset very unstructured.

Summarized, the dataset provides a large collection of documents with information about the author, the subreddit, and the content.

3.5. Preprocessing

Being user-generated text, the documents are very inconsistent in many ways and show a wide variety of different contents, ranging from well-written prose over character-based tables to nonsensical gibberish. For our purposes, we want to focus on documents containing well-written text and want to exclude any others. Before this is done in a filtering process, we perform various preprocessing operations on the data at hand:

1. Some comments consist mainly of URLs. Before determining whether these comments should be removed entirely, we remove the URLs to decide whether the remaining document contains enough text. Reddit supports limited formatting of content using a markdown-like syntax, including hyperlinks that have the following syntax:

```
[link label](link url)
```

We replace all markdown links with their *link label* and remove the information regarding the URL.

2. We replace any remaining URLs that were not formatted as markdown links by the term <URL>.
3. The Reddit syntax supports the citation of other text. As the resulting datasets are intended to be used for authorship attribution, we consider

citations not to be content of the original author and remove any lines marked as such (i.e., lines that start with '`>`').

4. for each message in the dataset, we determine its language using the *langdetect*¹⁰ library.

3.6. Filtering

Reddit comments contain many different types of text (e.g., ASCII-art, tables, etc.), which may not be of interest for any NLP task. In this work, we focus on generating text datasets containing plain written text, suitable for NLP tasks like authorship attribution or topic detection. This means that comments containing non-plain text have to be excluded. Examples of messages that are excluded by each of the presented measures can be found in Appendix B.2. We utilize simple textual features to filter out the following types of comments:

- When a user account is deleted on Reddit, the author field of messages by that user is set to `[deleted]`, and no information about the specific user can be retrieved. We drop comments with such an author field, as they are of no use for many tasks such as authorship attribution.
- Comments which are less than t_C characters long are discarded. This increases the expressiveness of the content by dropping short and often meaningless messages, helps the language detection to work more accurately, and reduces the file size of the dataset. This is a substantial benefit, as the entire dataset is over 2TB large and requires large amounts of time to process.
- If a comment does not have at least t_W words remaining after removing all punctuation marks, it is excluded. Otherwise, it is kept with its original punctuation. This step helps to filter comments consisting of ASCII-art or tables.
- After transforming the content to lower case, comments which have less than t_V *distinct* words are discarded. This helps to remove messages consisting of only a few words, repeated over and over. The casing of the remaining messages is left untouched.

¹⁰<https://pypi.org/project/langdetect/>, version 1.0.7 was used in this thesis.

- The language detection tool that we use estimates the probability of multiple languages for a text. By setting a threshold of t_L , we only keep messages which can be assigned a language with high confidence.

While these methods are simple, they are quickly calculated, can be applied universally to comments of all languages and topics, and manual inspections of the resulting datasets presented in Section 3.9 show that no unwanted content is left in the generated datasets. For our experiments, we used values $t_C=1,000$, $t_W=50$, $t_V=20$, $t_L=0.99$. All of these steps are configurable in the provided scripts, enabling both customizable datasets as well as reliable reproduction of a composed dataset.

From the initial three billion messages, 50 million messages remain after these processing steps. The according statistics are shown in Table 3.2. The filtered version of the entire dataset is the starting point for composing all datasets described in the remainder of this section.

3.7. Dataset Compilation

The comments in the dataset feature three categorical fields that can be used for text classification purposes: authorship, language, and subreddit. The authorship and subreddit fields can be used as target y for classification tasks, enabling the generation of datasets for authorship attribution and topic detection, respectively.

In theory, the presented setup can also be used to create *language detection* datasets by setting the $y = \text{language}$. However, the language of the documents is itself calculated automatically, and hence is not a solid ground truth.

Often, the large number of comments has to be limited to match custom requirements. By providing a minimal message length c as well as a minimal document number m per target, datasets of different sizes can be created, ranging from two to thousands of target classes, which can be used by large-scale models [44, 64]. Although such a limitation was already applied in the previous step, a more restrictive value may be chosen at this point, yielding longer texts.

Additionally, each of the three classification target fields (author, language, and subreddit) can be restricted to specific values by setting the respective restrictions denoted as R_A , R_L , and R_S , respectively. For example, when setting

y	R_A	R_L	R_S	resulting dataset
author	—	—	—	mixed language, mixed topic authorship attribution (AA)
author	—	{en}	—	single language, mixed topic AA
author	—	{en}	{/r/ama}	single language, single topic AA
author	{u1,u2}	{en,de}	—	mixed language, mixed topic AA for 2 specific users u1, u2
subreddit	—	—	{/r/ama, /r/politics}	mixed topic detection for 2 specific topics /r/ama, /r/politics

Table 3.3.: Examples of limiting fields and resulting datasets. y denotes the classification target, R_A , R_L and R_S stand for the restrictions (i.e., possible values) for *author*, *language* and *subreddit*, respectively.

$R_S = \{/r/AskReddit\}$, the resulting dataset will only contain documents that origin from the */r/AskReddit* subreddit. Table 3.3 shows how various values for these restrictions lead to different datasets. When providing no such restriction, the values for the respective field will include different, *mixed* values. For example, by setting the classification target $y = \text{author}$ with no restriction, documents of different topics and languages are collected for each author, without grouping them. This notation of *mixed* does therefore not refer to any cross-domain division of the data but rather states that the respective field contains different values, as no distinction between training and testing data is made at this point. While this is an undesired property for many use cases, models using domain-independent features (e.g., [58]) are still able to use these datasets.

Cross-domain datasets can be created by specifying an additional grouping field G . Thereby, only those target values are included in the result set if they feature at least m comments for every possible value in the grouping field. For example, if $y = \text{author}$, $G = \text{language}$ and $m = 5$, only those authors are kept who have written at least 5 comments in *every* language available. In most cases, this means that G must be limited by setting the respective restriction R to ensure that the intersection yields any results. For the previous example, setting $R_L = \{\text{en, de}\}$ relaxes the restrictions to only include those authors that have written at least 5 comments in both German and English.

Further examples of possible configurations and the resulting datasets are displayed in Table 3.4.

Furthermore, by tweaking the constraints c and m , different sizes of datasets can be created. Table 3.5a shows that by using small values for m , the dataset

y	G	example use-case
author	—	mono/mixed authorship attribution
author	subreddit	cross-topic/genre AA
author	language	cross-language AA
subreddit	—	topic detection (TD), genre detection (GD)
subreddit	language	cross-language topic detection (CLTD)

Table 3.4.: Examples for different dataset types generated by selecting different values for G . All restrictions from Table 3.3 can still be applied.

m	c				m	c		
	1,000	2,000	4,000	6,000		1,000	2,000	4,000
10	61,251	7,543	467	87	10	4,550	403	17
30	13,323	1,085	54	12	30	743	45	0
50	5,970	382	23	3	50	292	12	0
70	3,391	195	14	1	70	139	5	0

(a) Single-Topic AA, $R_S = \{/r/AskReddit\}$, $G = \{\}$ (b) Cross-Topic AA, $G = \text{subreddit}$,
 $R_S = \{/r/worldnews, /r/AskReddit\}$

Table 3.5.: Effect of minimal comment length c and minimal document count m on generated dataset size in terms of number of resulting target classes, for $y = \text{author}$, $R_A = \{\}$ and $R_L = \{en\}$.

allows the generation of large single-domain datasets with tens of thousands of authors, and even cross-domain datasets (Table 3.5b) with thousands of authors.

It is important to note that all steps described in this section are deterministic, and running the composition process with the same parameters will always result in the same dataset.

3.8. Cross-Topic/Genre Datasets

Grouping the comments by subreddit enables the generation of cross-topic datasets. It should be noted that while the authorship and language labels are clearly defined, the *subreddit* field should be used more carefully:

- Some subreddits like */r/worldnews* mostly comprise coherent discussions about a single topic whereas other subreddits like */r/AskReddit* are more diverse, where posts, and by extension, the comments, can have very different subjects.
- Many subreddits have rules regarding the content that can be posted, and moderators enforce them by filtering the content. However, in many cases, these rules only regard the original *post*, and less often extend to the comments belonging to that post. Often, discussions in the comment section of a post will diverge from the originally posted subject, to a point where the subreddit of the post becomes completely unrelated.

Existing datasets feature similar properties, where some target classes are more similar to others. For example, in the *Guardian*-dataset [86], which is a widely used cross-topic and cross-domain dataset, some topics (*Politics, World, UK*) are more similar to each other from a content-based point of view than others (*Books, Society*).

Depending on the subreddit, the resulting dataset might also consist of different text genres. For example, in the subreddit */r/WritingPrompts*, users often will post a short text with a creative idea that functions as a seed, and other users continue the story by writing appropriate passages in the comment section. These prose texts represent a different type of text than many other subreddits, and if */r/WritingPrompts* $\in R_S$, a cross-genre dataset can be composed.

While this does not invalidate utilizing subreddits as a restriction as a whole, it demonstrates that a certain amount of Reddit-specific domain knowledge is required to be able to compare results. More information about the content could be extracted using topic modelling techniques such as latent Dirichlet allocation or non-negative matrix factorization, but this step has not been performed on the entirety of the data collection, as it is time-consuming.

3.9. Cross-Language Datasets

Although many different types of datasets can be generated using our framework, we dedicate our attention in this section to the case of cross-language datasets, as these are underrepresented in literature, and often only translated versions of the original texts are used. In this section, a brief qualitative analysis of the comments is performed to better understand which languages can be used for creating cross-language datasets. In Table 3.6, the global distribu-

language	comments	% of collection
English	49,964,620	98.808%
Spanish	81,162	0.160%
German	79,969	0.158%
French	74,333	0.147%
Portuguese	61,386	0.122%

Table 3.6.: Most common languages used in the comments.

language	subreddit	description	comments	% of lang.
English	<i>/r/AskReddit</i>	general Q&A	3,787,110	7.6%
	<i>/r/politics</i>	politics	1,234,722	2.5%
	<i>/r/worldnews</i>	world news	738,144	1.5%
Spanish	<i>/r/podemos</i>	political party	81,162	55.1%
	<i>/r/argentina</i>	country	28,200	19.2%
	<i>/r/mexico</i>	country	17,188	11.7%
German	<i>/r/de</i>	country	44,835	56.1%
	<i>/r/rocketbeans</i>	media	8,533	10.7%
	<i>/r/Austria</i>	country	5,408	6.7%
French	<i>/r/france</i>	country	49,520	66.6%
	<i>/r/Quebec</i>	country / area	14,947	20.4%
	<i>/r/montreal</i>	country / area	1,519	2.0%
Portuguese	<i>/r/brasil</i>	country	31,830	51.8%
	<i>/r/portugal</i>	country	22,063	35.9%
	<i>/r/PremeiraLiga</i>	soccer	980	1.6%

Table 3.7.: Top three subreddits with the most comments for selected languages. The last column describes the subreddit’s fraction of all posts in the respective language.

tion of languages across all comments is shown. It confirms the intuition that English is by far the predominant language used in Reddit comments. It also demonstrates that for non-English languages, the subreddits concerning the countries where the respective language is spoken are among the biggest on Reddit.

For the five most popular languages on Reddit (English, Spanish, German, French, and Portuguese), we analyzed the distribution of these languages across different subreddits, shown in Table 3.7. In many cases, the subreddit with the most comments relates to the nationality of that language. In

(i) Aggregated Metrics

metric	definition	description
$dln_{w/c}$	$\{\#(d)_{w/c} \mid d \in D\}$	length of all documents in D .
$sln_{w/c}$	$\{\#(s)_{w/c} \mid s \in d, d \in D\}$	length of all sentences in D .
dpa	$\{ \mathbf{d}_a \mid a \in A_D\}$	documents per author.
$imb(a)$	$\sigma(\{\#(d) \mid d \in \mathbf{d}_a\})$	author imbalance: standard deviation of document lengths of author a
imb	$\{imb(a) \mid a \in A_D\}$	author imbalance of all authors

cases where the respective language is the native language of multiple countries (e.g., Portuguese in Portugal and Brasil), the respective distribution of subreddits shows related results.

In theory, this information can help to construct datasets based on subreddits to analyze different language dialects. For example, given the subreddits from Table 3.7, two different types of cross-language corpora can be created by setting $R_S = \{/r/brasil, /r/AskReddit\}$ or $R_S = \{/r/portugal, /r/AskReddit\}$, respectively. However, this approach is not analyzed in detail in this thesis and is instead left open for future research.

Table 3.8 shows different cross-language corpora created by varying only the R_L restriction parameter, while leaving the other parameters y =author, G =language, c =3,000 and m =20 are constant. Interestingly, the sizes of the created corpora no longer correlate with the distribution of the languages in total. For example, while there are more Spanish comments than French ones (cf. Table 3.6), there seem to be more French users writing in English than Spanish ones. In total, the French-English dataset is the largest according to multiple metrics, including the total number of documents, the number of authors, and also the average number of documents for each author.

It is also noticeable that Reddit is an internet platform with mainly English and European speaking users. For languages outside of this set of languages, the sparsity of bilingual users writing also in English makes it difficult to find enough data to compose comparable datasets. For example, when using $R_L = \{en, ar\}$ in combination with c =3,000 and m =20, no documents remain after filtering. Only when reducing the c and m restrictions (cf. Table 3.8), suitable datasets can be obtained, which then are no longer directly comparable to the other datasets due to the different sizes.

dataset	R_L^*	$ A_D $	$ D $	$\overline{\text{dpa}}$		$\min_{a \in D_A} \text{dpa}$		$\sigma(\text{dpa})$	$\overline{\text{imb}}$	$\overline{\text{dln}_c}$
R1-DE	de	28	4,087	84 _{EN}	62 _{DE}	21 _{EN}	20 _{DE}	129	269	3,733
R2-ES	es	20	4,450	118 _{EN}	52 _{ES}	20 _{EN}	21 _{ES}	204	233	3,125
R3-PT	pt	37	4,481	69 _{EN}	52 _{PT}	20 _{EN}	20 _{PT}	83	227	2,995
R4-NL	nl	11	2,410	155 _{EN}	32 _{NL}	20 _{EN}	20 _{NL}	137	266	3,231
R5-FR	fr	45	10,131	103 _{EN}	61 _{FR}	21 _{EN}	20 _{FR}	173	257	3,088

Table 3.8.: Sizes of different cross-language AA datasets. *All R_L restrictions are combined with ‘en’. For all entries $c = 3,000$ and $m = 20$.

The column named “ $\overline{\text{dpa}}$ ” contains the average number of documents per author for each of the two respective languages of the dataset, while the column “ $\min \text{dpa}$ ” shows the minimum of that value. Comparing the two columns shows that the datasets are not balanced in the sense that on average, users write significantly more English documents than in the respective other language. Additionally, the column σdpa displays the standard deviation of the numbers of documents written by each author and demonstrates that each dataset contains some users that write much, while others feature only a few documents.

3.10. Conclusion

In this section, the need for evaluation resources containing multilingual authors is explained, motivating the presented approach to compose different datasets based on a large pool of social media comments using the Reddit platform. Using this technique, datasets for cross-topic, -genre, or -language experiments can be created by filtering the appropriate documents from the large collection. Various parameters in this processing allow for fine-grained control over the size and characteristics of the respective datasets.

More specifically, this chapter answers the first research question: *How can datasets be obtained that are suitable for CLAA?* by showing that using social media comments can be used to compose datasets that fulfill all requirements for cross-language authorship attribution, and the approach can be used to compose datasets in multiple language combinations.

In the remainder of this thesis, the datasets listed in Table 3.8 are used in the various evaluation experiments. Thereby, they will be referred to using the first column of the table (e.g., R1-DE).

As stated at the beginning of this section, the data for composing these datasets was gathered in 2018. Reddit has since gained many users and comments, and some of the relationships between languages may have changed. Including the latest data dumps from Reddit is necessary for future analyses of this topic. This may yield both larger datasets for the language combinations presented in this section, as well as provide novel language combinations that have been previously too sparse to use the presented framework to compose datasets.

DT-Grams: Dependency-Graph Substructures¹¹

Section 2.2 presented several methods to extract information from text that is frequently used in various types of NLP research. Later, in Chapter 3, a foundation for cross-language authorship attribution experiments was provided with a framework to construct cross-language datasets from social media comments. This chapter combines these areas and shows how information embedded in the grammar of documents can be used as a source for additional information and introduces DT-grams as a cornerstone and one of the main contributions of this thesis, developed with answering the second research question in mind: *Which language-independent syntax-based features are a viable choice for a classification feature for CLAA?*

Thereby, this chapter will first discuss grammar parse structures before explaining substructure extraction to provide machine learning features from the parse trees. Then, DT-grams are introduced along with the language-independent universal grammar features that they incorporate, before providing an outlook on how these features can be used in different machine learning models.

Using the datasets previously presented in Chapter 3, an in-depth evaluation of DT-grams is performed, including both the fine-tuning of the parameters that DT-grams provide, as well as experiments and comparisons involving existing state-of-the-art models for text classification.

¹¹Results and contents of this chapter are based on and partially reused from my papers:

- Benjamin Murauer and Günther Specht: *DT-grams: Structured Dependency Grammar Stylometry for Cross-Language Authorship Attribution*. In Proceedings of the 32nd GI-Workshop Grundlagen von Datenbanksysteme (GvDB'21) . 2022
- Benjamin Murauer and Günther Specht: *Small-Scale Cross-Language Authorship Attribution on Social Media Comments*. In Proceedings of the 4th Workshop on Technologies for MT of Low Resource Languages (LoResMT2021), pages 11-19. 2021

4.1. Introduction

One of the main problems of cross-language text classification is the choice of features that are extracted from the documents. Section 2.2 explains the basic concepts of which properties of a text can be extracted and it becomes clear that many of the presented approaches are strongly affected by a cross-language dataset. For example, comparing the frequency of words is no longer a viable option when the train and test documents don't share a common vocabulary. Similarly, using character-based features is limited to language combinations using the same alphabet, and, for example, is of no use for comparing English and Korean documents.

Therefore, CLAA models must either use features that are suitable for a cross-language setup, or find other ways to overcome this obstacle. The following section is dedicated to the introduction of the DT-gram feature, following the former approach. At the end of this chapter, evaluation experiments are included that demonstrate how the DT-grams compare to different methods following the latter approach by using machine translation to overcome the language gap in the dataset.

The basic concept of DT-grams can be broken down into two key ideas: By leveraging language-independent representations of words using universal POS tags, features suitable for cross-language text classification can be extracted from the text. Additionally, the POS tags are grouped according to distance measures within a dependency graph to create syntactic n -grams.

4.2. Related Work

Using POS tags is a widely used feature in many different text classification tasks, including authorship analysis tasks like authorship profiling [95], translation detection [63] or authorship attribution [5, 83]. As a close match to the work presented in this thesis, Bogdanova et al. [5] also use language-independent *universal* POS tags for authorship attribution (the nature of universal POS tags will be explained in more detail in the following sections). However, in their work, they compose n -grams of these tags, thus using the original word order of the sentences. In this chapter, we demonstrate how using word neighborhoods determined by dependency graphs instead of using regular n -grams is able to improve this concept.

Syntactic features are also a feature that has been included in several NLP studies. Sidorov et al. [83] calculate syntactic n -grams by considering words as neighbors that are in a parent-child relationship in a constituency or dependency parse tree. Furthermore, they incorporate multiple representations of the words themselves, including using their POS tag as well as their original form. The most important differences to the work presented in this thesis is that their experimentation scenario was focused on mono-language datasets, and that their concept is limited to the direct ancestor relationships of the nodes in the parse trees. These limitations are shared by the work of Zhang et al. [101], which also utilize similar features for authorship attribution, but calculate embeddings of these syntactic n -grams using convolutional neural networks.

Tschuggnall et al. have used the constituency parse trees of sentences as an additional source of structural syntax information for various tasks including authorship attribution [94], profiling [95] or plagiarism detection [91, 92, 93]. While in their work they use (mono-lingual) constituency trees and language-dependent POS tags to calculate this feature, the work presented in this thesis focuses on extending this approach towards language-independent classification using dependency grammar and language-independent, universal POS tags.

4.3. DT-Grams: Dependency Tree Substructures

By parsing the sentences and obtaining their dependency graph, the task of comparing sentences has transformed into comparing graph structures. Augsten et al. [3] introduced the pq -gram distance as an approximated measure for tree similarities, which has been used by Tschuggnall et al. [91, 94, 95] with constituency trees for several linguistic problem settings. In this work, similar approaches are used, but additional strategies and parameters are experimented with. For the remainder of this section, the dependency graph structures (e.g., Figure 2.3b) are interpreted as tree structures by selecting the root of the dependency chain as the tree root node.

The basic idea of this approach is to count how many substructures of a tree are occurring in both compared trees. Therefore, a stencil shape σ is required in addition to the trees themselves. This stencil determines which nodes of the tree are considered to belong to the same substructure, based on the distance of the nodes to their respective ancestors and siblings. When comparing the procedure to word n -grams in traditional text, the stencil can be interpreted as a window of n words that is moved across the text document.

In the case of a tree structure as the underlying source of data, this stencil can have more than one dimension. Figure 4.1 depicts four stencils that have different sizes (how many ancestors/siblings are considered) and shapes (in which configuration the ancestors and siblings are considered). In the remainder of this thesis, both the shape σ as well as its respective dimensions δ_{sib} (number of selected siblings) and δ_{anc} (number of selected ancestors) are hyperparameters that can be tuned for a specific downstream task. This in itself is a novel addition to the experiment setup, as Tschuggnall et al. use one fixed stencil shape recommended by Augsten et al. for a different type of similarity measurement task.

In the remainder of this thesis, the four stencil shapes shown in Figure 4.1 are used as candidates after performing preliminary experiments with a larger set of shapes, which are listed in Appendix A. The candidates are based upon the results of the bachelor thesis by Philipp Pobitzer [71].

Figure 4.2 shows how the stencil is moved across a tree. Thereby, the stencil is placed on top of the root node (cf. step “a” of the figure). In some cases, the stencil does not fit onto the tree structure. For example, in the example shown in Figure 4.2, the node 1 does not have any children, but the stencil is extending to their theoretical positions. Then, the “free” places are filled with a wildcard element $*$. At each position, the stencil counts an instance of a DT-gram by concatenating all nodes (or wildcards) currently filled in the stencil. For example, the stencil at position 4. of the figure will extract one instance of the DT-gram $0-2-3-4-*$. Then, the stencil is moved to the next available spot.

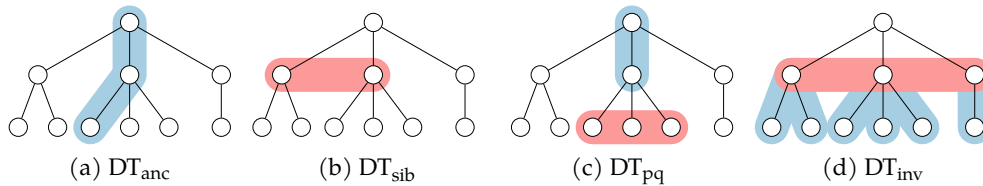


Figure 4.1.: Selected stencil shapes for extracting tree substructures. The number of ancestors (blue) and siblings (red) considered by each stencil is a hyperparameter that can be tweaked by a downstream task.

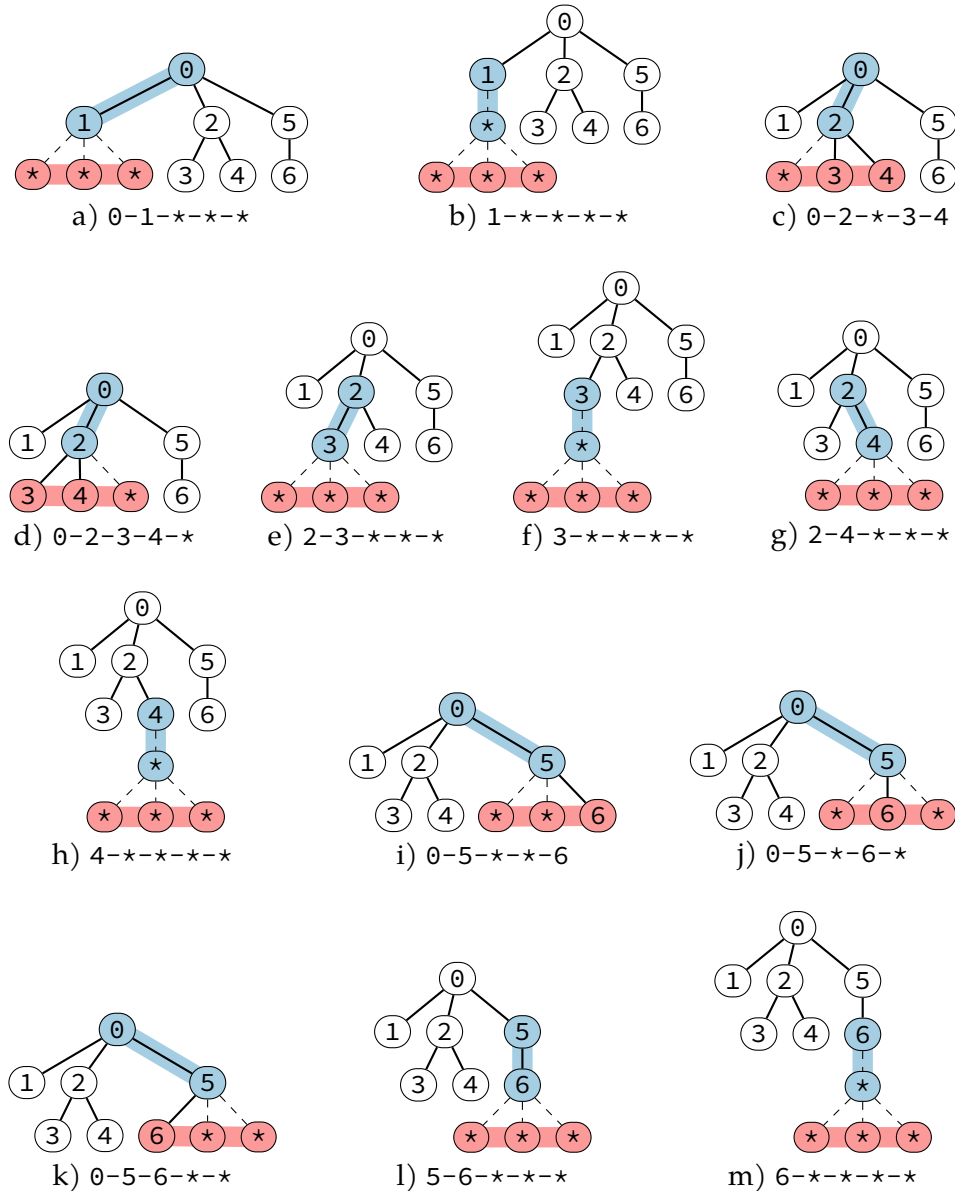


Figure 4.2.: Movement steps of the stencil with parameters set to $\sigma = pq$, $\delta_{anc}=2$ and $\delta_{sib}=3$. The terms on the right side denote the DT-gram that is produced by the stencil at the respective position. Empty positions in the stencil are filled with a placeholder element $*$.

The multiset of all instances that is extracted from a tree t is denoted as $dt(t)$. For the example in Figure 4.2, it contains 13 elements:

$$dt(t) = \{ 0-1-*-**, 1-*-**, 0-2-*-3-4, 0-2-3-4-*, \\ 2-3-*-**, 3-*-**, 2-4-*-**, 4-*-**, \\ 0-5-*-**6, 0-5-*-6-*, 0-5-6-*-**, 5-6-*-**, \\ 6-*-** \}$$

Depending on the number of dimensions that a stencil has, there are multiple strategies on how the stencil can be moved across the tree. For example, after position 1. of Figure 4.2, the stencil can be moved in multiple ways:

- depth-first: moving it down one node towards position 2, or
- breadth-first: moving it to the right one node towards position 4.

Models that use the frequency of DT-grams are not affected by this difference, but models that use DT-gram sequences from the grammar structures should be aware of this, and optionally choose the best strategy by tuning it as a hyperparameter.

Similarly, the order in which the respective nodes of the tree structure are used in the serialized DT-gram must be defined in order to produce stable features, especially for overlapping stencil shapes. In this thesis, the extraction strategy follows the suggestions of Austen et al., who propose a preorder method. While the choice of this strategy does not influence the number of features extracted from the tree, it is important that it remains the same for all extraction procedures within an experiment to ensure that similar substructures can be compared as such.

In the previous section, both the leaf nodes of the constituency tree and the nodes of the dependency graph are represented by their original words. This enables a more direct comparison of the extracted DT-grams to the commonly used word n -gram feature in the sense that both feature families count the occurrences of groups of words that can be considered neighbors, either in the original word order or in a grammatical context. Therefore, the DT-grams also share a common downside of the word n -grams: the extracted features are highly sparse and the feature space increases dramatically with the size of the extraction stencil.

This can be mitigated by not using the original form of the word to represent the corresponding node in the dependency graph or constituency tree, but rather using a more generic representation like the word's lemma or POS tag. The latter approach also has a benefit for evaluation purposes, as the features are inherently unable to capture content, and are therefore highly suited for analyzing the *style* of an author. Tschuggnall et al. [91, 92] exploit this for intrinsic plagiarism detection, and show that the change of style of a text alone is able to indicate plagiarism. The following section demonstrates that by using an appropriate representation for each word, language-independent features can be generated.

4.4. Language-Independent Grammar Features

tag	name	example
VAFIN	finite auxiliary verb	sie ist gekommen
VAIMP	imperative of auxiliary	sei still!
VAINF	infinitive of auxiliary	er wird es gesehen haben
VAPP	past participle of auxiliary	sie ist es gewesen
VMFIN	finite modal verb	sie will kommen
VMINF	infinitive of modal	er hat es sehen müssen
VMPP	past participle of auxiliary	sie hat es gekonnt
VVFIN	finite full verb	sie ist gekommen
VVIMP	imperative of full verb	bleibt da!
VVINFINF	infinitive of full verb	er wird es sehen
VVIZU	infinitive with incorporated zu	sie versprach aufzuhören
VVPP	past participle of full verb	sie ist gekommen

(a) German POS tags of verbs used in the TIGER corpus [7]

tag	name	example
VB	verb, base form	take
VBD	verb, past tense	took
VBG	verb, gerund/present participle	taking
VBN	verb, past participle	taken
VBP	verb, sing. present, non-3rd	take
VBZ	verb, 3rd person sing. present	takes

(b) English POS tags of verbs from the Penn tree bank [56]

Table 4.1.: Different languages have different grammatical features.

4.4. Language-Independent Grammar Features

The introduction of the additional difficulty of *cross-language* classification renders the direct use of POS tags impossible, as different languages feature different tags and grammar parsing rules. For example, Table 4.1 shows the different tags of verbs in German and English. It is clear that both languages use different forms of verbs and therefore have different means of tagging them.

One way to circumvent this problem is to utilize *universal* grammatical information mappings [15, 67, 66]. Thereby, the language-specific POS tags are mapped to a universal, language-agnostic space. By definition, this causes a loss of fidelity in these measures but allows direct comparisons between documents written in different languages. Concretely, all 12 German and 6 English POS language-specific tags displayed in Table 4.1 are mapped to either “AUX”

or “VERB” in the universal POS space. At the time of writing this thesis, the full list of universal POS tags has 17 entries and can be found in the appendix in Table D.1.

Similarly, a universal mapping of dependency grammar labels is also available by the same research initiative. The full list of 37 universal dependencies can be found in Table D.2. An elaborate documentation of the development and classification of these universal resources can be found on the project’s homepage¹².

By parsing documents in both languages and utilizing the universal POS tags of each word as the representation of the word, traditional n -grams can be computed that can be used for cross-language text classification. The universal POS tags of the previously used sentence “I have been trying to reach you” are:

	I	have	been	trying	to	reach	you	.	
Eng. tag	PRP	VBP	VBN	VBG	TO	VB	PRP	.	
Univ. tag	PRON	AUX	AUX	VERB	PART	VERB	PRON	PUNCT	

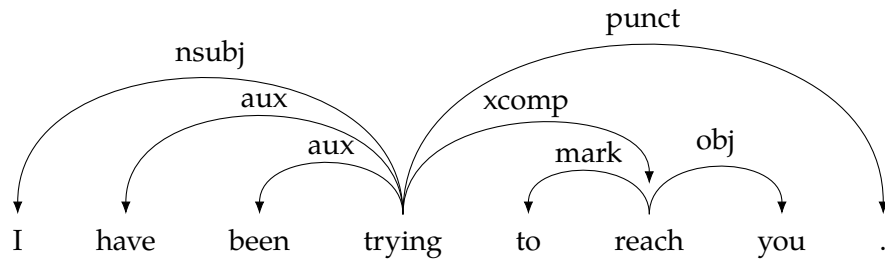
Applying the process of extracting 3-grams (cf. Figure 2.2) using the universal pos tags of the same sentence results in a set of POS tag 3-grams starting with PRON-AUX-AUX, AUX-AUX-VERB, AUX-VERB-PART, etc.

These tags can then be interpreted as tokens for the remaining machine learning steps, for example by counting their frequency or by using them to compute a lower-dimensional embedding (cf. Section 2.2). Note that by using universal POS tags, the lower number of available tags drastically reduces the feature space of possible n -grams depending on the size of n . Table 4.2 shows some reference numbers for this feature space size using five of the Reddit datasets introduced in Chapter 3, and clearly demonstrates the difference in the number of distinct language-dependent and universal POS tag 3-grams.

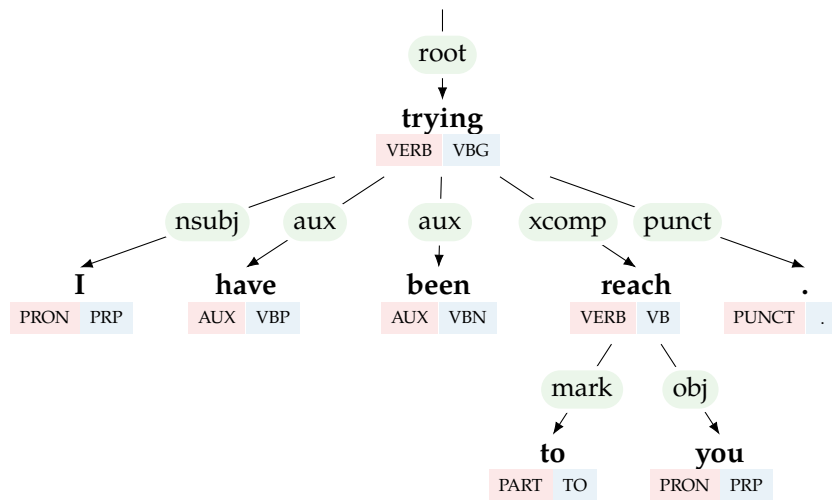
Similarly, the nodes of grammar parse structures can also be represented by universal POS tags.

Figure 4.3 displays various representations of the sentence “I have been trying to reach you.”. In Figure 4.3b, the original dependency graph (4.3a) is displayed as a tree structure, where each node is a word (including its English and universal POS tag) and each edge is labeled with the dependency

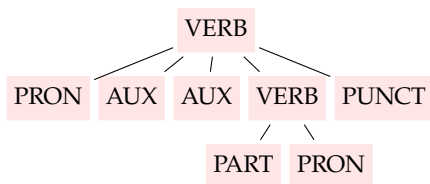
¹²<https://universaldependencies.org>



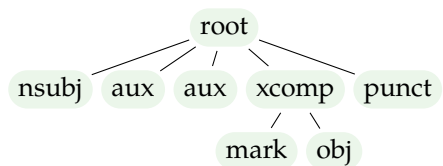
(a) Dependency graph of the sentence "I have been trying to reach you".



(b) Dependency graph displayed as tree structure including dependency relationship names, universal POS tags (red) and English POS tags (blue).



(c) Tree with universal POS tags as nodes.



(d) Tree with dependency labels as nodes.

Figure 4.3.: Different representation of the sentence "I have been trying to reach you". The relationships of the dependency graph (a) can be combined with English or universal POS tag information, yielding different representations of the sentence (c, d).

POS tags	n	Reddit dataset				
		R1-DE	R2-ES	R3-PT	R4-NL	R5-FR
Universal	1	17	17	17	17	17
	2	287	289	289	289	289
	3	3,718	3,905	3,970	3,816	4,271
Language-Specific	1	101	64	50	324	52
	2	3,215	2,069	1,875	7,253	1,962
	3	37,881	27,569	26,153	56,714	30,674

Table 4.2.: Number of distinct POS tags n -grams for universal and language-specific POS tags in the Reddit dataset.

relationship. From this structure, different sub trees can be extracted by selecting the different POS tags or dependency labels (4.3c, 4.3d).

The choice of which part of the node in the graph is a parameter that is named η in the remainder of this thesis. It can be optimized for downstream tasks, but may also influence the evaluation characteristics of a problem. For example, using either language-dependent or universal POS tags renders the feature inherently unaffected by the topic of a text, since no semantic content remains in the document. This can be useful for determining whether or not an evaluation result is dependent on the topical difference between training and testing documents, or whether observations can be attributed to an author's *style* exclusively. Inversely, using the original word as representation allows comparisons between the classification performance of grammar-based word neighborhoods and the original word order provided by the author.

In principle, both constituency trees and dependency graphs can be used for extracting tree or graph substructures and using language-independent features for representing the nodes. However, dependency grammar is used in this thesis for two reasons:

1. The order of words within the sentence does not influence the dependency relationships but does influence the constituents of sentences. For example, the two sentences "I left singing." and "Singing, I left" have two different constituency trees, but show the same word dependencies. Our argument is that since different languages have different orders in which certain words are placed within sentences, the invariant word dependencies help to identify patterns better than constituency grammar.
2. There is no broadly accepted universal constituency model. This means that while documents could be parsed in the respective languages' con-

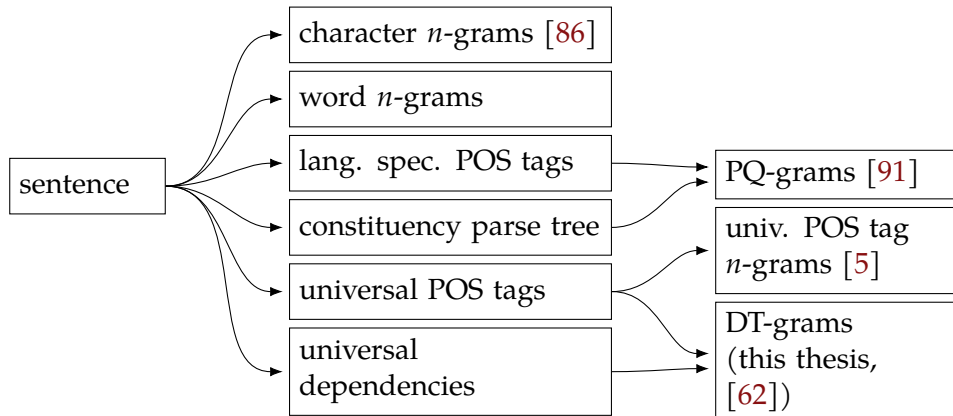


Figure 4.4.: Summary of grammar features described in this chapter.

stituency model, the representation of the nodes can't be mapped to a language-independent universal model. Dependency grammar, on the other hand, represents a simpler concept, and a universal, language-independent mapping has been developed. Parsers are widely available for a wide variety of languages [66] while constituency parsers are scarce, especially for low-resource languages.

Hence, the remainder of this thesis uses universal dependencies to provide the grammatical structure which is used to extract the features. The name of the features represents this choice, where the DT in DT-grams stands for *Dependency Tree*.

To this point, several methods for extracting different types of grammatical features have been presented. Before this chapter continues to explain the different possibilities of how these features can be used in machine learning setups, Figure 4.4 summarizes the most important grammatical features for authorship analysis discussed in this chapter, and which of those features are used by related work in this field.

4.4.1. Summary: DT-grams

Before the next section introduces experiments to evaluate different properties of the DT-gram feature, a short summary of the DT-gram feature can be explained as follows:

- a parser is used to determine grammatical structure of sentences

- a representation of each word is selected (η)
- substructures of the grammar trees are extracted, where parameters determine the shape (σ) and size ($\delta_{anc}, \delta_{sib}$) of the substructures

<i>i</i> DT-gram parameters		
	description	possible values
σ	shape of the stencil	$DT_{anc}, DT_{sib}, DT_{pq}, DT_{inv}$
δ_{anc}	ancestors included in shape	1, 2, 3, ...
δ_{sib}	siblings included in shape	1, 2, 3, ...
η	word representations	universal POS tag, language specific POS tag, original word, lemma, ...

These steps transform a text document into a stream of DT-grams, which can then be used in a machine learning pipeline in different ways.

4.5. Features and Models Using DT-grams

In this section, the models that use the previously introduced DT-grams feature will be explained in more detail. The models that are used for baseline comparisons in the experiments in this chapter are discussed in the foundations in Chapter 2.

4.5.1. Frequencies of DT-grams

The most straightforward way to use DT-grams as features is to count their frequency analogous to counting words in word-1-grams. This is also semantically relatable to word- n -grams, as it counts how often a set of n words occur “together” in a document. However, the DT-grams are calculated on a per-sentence basis, which enables two different approaches to how the classification of the original document can be achieved. Firstly, combining the DT-grams of all sentences of one document d into one large set of DT-grams:

$$A(d) = A \left(\bigcup_{s \in d} \text{DT}(s) \right) \quad (4.2)$$

where $A(X)$ is the predicted authorship of X determined by some model and $\text{DT}(s)$ is the set of all DT-grams extracted from sentence s . Secondly, by splitting the original documents into their constituent sentences and performing a majority-voting to determine the authorship of the entire original document:

$$A(d) = \arg \max_{a \in A_D} \left(\sum_{s \in d} A(\text{DT}(s)) = a \right) \quad (4.3)$$

In either case, additional normalization techniques can be applied, including computing the tf/idf norm or only using the most frequent x features. The sentence majority voting (Equation 4.3) is therefore, strictly seen, a sentence classification method. Since we expect the single sentences to contain fewer diverse DT-grams as they are much shorter than the documents, the unifying approach as listed in Equation 4.2 is used in the remainder of this thesis.

Using frequencies of DT-grams provides similar benefits and disadvantages as the traditional bag-of-words approach:

Benefits

- fast computation
- intuitive interpretability

Disadvantages

- produces sparse features - may not work with some models
- high frequency does not mean high importance

4.5.2. DT-gram sequences

By interpreting the collection of DT-grams that are extracted from a document as a continuous stream of tokens, different machine learning models can be utilized, like recurrent neural network models [103, 102, 48] or hidden Markov models [100, 32]. Note that as stated in Section 4.3, the order in which the tree substructures are extracted from the dependency graph is important in this case, and should be the same for all documents in an experiment.

4.5.3. Kernel Methods for DT-Grams

Yet another way of using the similarity between documents as a machine learning feature is to use measurements in a kernel function of a model that only uses the inner product of feature vectors. A typical example of such a model is the support vector machine, which is explained in Section 2.4.1. The work by Ionescu et al. uses the presented kernels with *character*-based subsequences of strings (i.e., they extract character p -grams), but the idea behind the kernels is applicable to a much more general set of document features. In this thesis, the approach is used to measure document similarities based on the co-occurrence of the previously presented DT-gram features. The adopted versions of the kernels no longer have the parameter p , but instead have the parameters σ (shape) and δ (dimensions) of the DT-grams (cf. Section 4.3) and iterate over the set of DT-grams in the entire document set D :

$$\begin{aligned}
 k_{\sigma,\delta}(x_i, x_j) &= \sum_{v \in \text{DT}(D)} \text{num}_v(dt(x_i)) \cdot \text{num}_v(dt(x_j)) \\
 k_{\sigma,\delta}^{0/1}(x_i, x_j) &= \sum_{v \in \text{DT}(D)} \text{in}_v(dt(x_i)) \cdot \text{in}_v(dt(x_j)) \\
 k_{\sigma,\delta}^{\cap}(x_i, x_j) &= \sum_{v \in \text{DT}(D)} \min(\text{num}_v(dt(x_i)), \text{num}_v(dt(x_j)))
 \end{aligned} \tag{4.4}$$

To prevent the length of a document influencing the kernel value, we implement the following normalization, as suggested by Ionescu et al. [33]:

$$\hat{k}(x_i, x_j) = \frac{k(x_i, x_j)}{\sqrt{k(x_i, x_i) \cdot k(x_j, x_j)}} \tag{4.5}$$

In the evaluation experiments, we use the kernels $\hat{k}_{\sigma,\delta}$, $\hat{k}_{\sigma,\delta}^{0/1}$ and $\hat{k}_{\sigma,\delta}^{\cap}$ with the support vector machine classification model. Thereby, the `scikit-learn`¹³ software library is used, which internally makes use of the `libsvm`¹⁴ implementation of the algorithm.

¹³<https://scikit-learn.org>

¹⁴<https://github.com/cjlin1/libsvm>

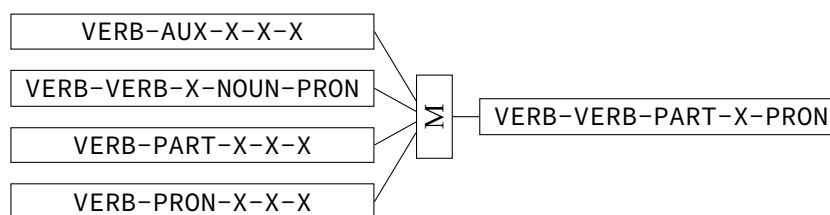


Figure 4.5.: Example of an embedding in the Word2Vec “continuous bag-of-words”-style using DT-grams, where M denotes the hidden layer containing the resulting embedding weights. Other techniques like GloVe can be applied analogously.

4.5.4. DT-gram Embeddings

In Section 2.3.2, the basic principles behind word- and document embeddings are explained. This concept can be easily extended by using a DT-gram as token and using the sequence of DT-grams as context. For example, the embedding strategy of Word2Vec using this approach is depicted in Figure 4.5 (but the principle applies to any embedding strategy).

In recent work, the output of large pre-trained language models has been used as a substitute for traditional embeddings, yielding a lower-dimensional representation of the respective words that can be used as input for further machine learning models. In order to map this approach to the concept of DT-grams, the language model must be pre-trained on appropriate data as well. We identify this as an interesting concept for future research, but implementing it requires vast amounts of data, which in this scenario, must be fully annotated including universal dependencies and universal POS tags. Even the largest treebanks are not comparable to the amounts of data that are required for such a task (e.g., the BERT model is trained on two datasets consisting of 800M and 2,500M words, respectively, compared to the 2.5M words of the widely used Penn Treebank corpus [56]). To the best of our knowledge, no such data collection exists.

4.6. Finding Optimal DT-gram Parameters

In this section, the influence of the parameters on the performance of cross-language authorship attribution is analyzed in detail. The first goal of the evaluation experiments is to determine good values for the DT-gram dimensions (δ_{anc} and δ_{sib}), the shape (σ) and the node representation (η). We also want to determine whether the best values for these parameters depend on

the language of the dataset, the classification method that is used, and each other. Table 4.3 summarizes the parameters that are tested in this section:

parameter	values
shape σ	$DT_{anc}, DT_{sib}, DT_{pq}, DT_{inv}$ (cf. Figure 4.1)
dimensions δ_{anc} and δ_{sib}	$\{1, 2, 3, 4\}^2$
node representation η	Univ. POS tag, univ. dependency role, both

Table 4.3.: DT-gram parameters analyzed for cross-language authorship attribution.

4.6.1. Experiment Data

To measure how the DT-gram parameters effect the authorship attribution performance, a wide range of experiments is conducted, with the aim of covering many different aspects of such a task. This includes multiple datasets and classification models operating on the DT-gram features, as well as different methods of how the DT-grams can be incorporated into a machine learning model.

As evaluation datasets, the Reddit datasets presented in Chapter 3 are used. However, in their unaltered state, the datasets are very unbalanced in multiple dimensions:

- Some datasets have more authors than others (cf. column $|A_D|$ of Table 3.8). For example, R5-FR contains 45 authors, while R4-NL contains only 11.
- Some authors have more documents than others (cf. column $\sigma(\text{dpa})$ of Table 3.8). For example, authors in the R2-ES dataset have written an average of 118 English documents, while authors in the R3-PT dataset have only 69.
- All authors have more English documents than the respective other language (cf. column $\overline{\text{dpa}}$ in Table 3.8)

While this is inherently realistic, it impedes direct comparison of different Reddit datasets and the performance of the DT-gram parameters. We therefore use reduced versions of each Reddit dataset for the evaluation experiments, where we limit the number of authors and training documents to obtain directly comparable results (the number of testing documents is not modified).

Algorithm 1 Selecting training and testing data from an unbalanced Reddit dataset Rx .

```

1:  $S \leftarrow \emptyset$ 
2:  $p_{it} \leftarrow 10$  ▷ number of iterations
3:  $p_{auth} \leftarrow 10$  ▷ number of authors
4:  $p_{docs} \leftarrow 10$  ▷ number of documents per author
5: for  $i = 0; i < p_{it}; i = i + 1$  do
6:    $A^* \leftarrow$  select  $p_{auth}$  random authors from  $Rx$ 
7:    $S_{train} \leftarrow \{d \mid a_d \in A^* \wedge l_d = l_{train}\}$ 
8:    $S_{train}^* \leftarrow$  select  $p_{docs}$  random documents from  $S_{train}$ 
9:    $S_{test} \leftarrow \{d \mid a_d \in A^* \wedge l_d = l_{test}\}$ 
10:   $S := S + (S_{train}^*, S_{test})$ 
11: end for
12: return  $S$  ▷  $S$  contains tuples of (train, test) documents

```

1. train with 10 randomly picked English documents for each of 10 randomly picked authors, and test on all non-English documents
2. train with 10 randomly picked non-English documents for each of 10 randomly picked authors, and test on all English documents

Additionally, the experiments are repeated 10 times for *each* configuration, so that the effect of sampling the authors and documents is minimized. Algorithm 1 shows how the documents for each experiment in the remainder of this section are selected.

4.6.2. Experiment Methods

In Section 4.5, several different methods of utilizing DT-grams as machine learning features have been presented. We employ four different machine learning approaches in this setup, which incorporate each of the presented strategies. Concretely, we use the following classification models for the experiments:

1. $\text{svm}_{\text{tf/idf}}$: a linear SVM using tf/idf normalized DT-gram frequencies
2. $\text{xgb}_{\text{tf/idf}}$: a extreme gradient boosting classifier [12] using tf/idf normalized DT-gram frequencies
3. svm_{emb} : a linear SVM using Doc2Vec DT-gram embeddings

parameter	values
embedding method	distributed memory, distributed bag-of-words
embedding dimensions	20, 100, 300
embedding epochs	20, 100
kernel methods	intersection, presence, spectrum

Table 4.4.: Parameters of the classification approaches used in the experiments.

4. svm_{kern} : a SVM using kernel methods

Thereby, the svm_{emb} and svm_{kern} approaches have some parameters themselves which are explored in this experiment. The values used are listed in Table 4.4.

The choice of the parameters that are to be determined by the following experiments influence each other: DT-grams of one shape may be more expressive using different node representations than using a different shape. This makes evaluating a single best result difficult. Instead, in this section, a comprehensive grid search is performed that combines all possible parameter combinations, datasets, and classification approaches.

Note that the results of all experiments that have the same configuration except for the iteration mitigating sampling bias (cf. line 5 of Algorithm 1) are averaged. For example, the result set has exactly ten entries for the experiment with the approach = $\text{svm}_{\text{tf/idf}}$, $\delta_{\text{anc}} = 2$, $\delta_{\text{sib}} = 3$, $\sigma = \text{DT}_{\text{pq}}$, $\eta = \text{u.POS}$ tags and uses R1-DE with German training data and English testing data. Likewise, for all other parameter combinations, the 10 results of that combination are averaged in the remainder of this section.

The resulting set of evaluations is vast and guarantees to find the best parameters in the search space, but it makes visualizing them more difficult, as listing all results is not feasible. Therefore, from this large collection of results, important aspects and aggregated results for each of the parameters in Table 4.3 are presented in the following sections.

4.6.3. Influence of DT-gram Shapes

The first aspect of the DT-grams features analyzed is the shape of the tree substructures used. Unlike the research performed by Tschuggnall et al. [91], who only used one fix-sized substructure, this thesis analyzes a broader vari-

4.6. Finding Optimal DT-gram Parameters

σ	R1-DE	R2-ES	R3-PT	R4-NL	R5-FR	mean
DT _{sib}	0.34	0.26	0.24	0.26	0.27	0.27
DT _{anc}	0.34	0.23	0.22	0.26	0.28	0.27
DT _{pq}	0.40	0.28	0.23	0.27	0.31	0.30
DT _{inv}	0.38	0.26	0.22	0.27	0.30	0.29

(a) Train with English, test with non-English documents.

σ	R1-DE	R2-ES	R3-PT	R4-NL	R5-FR	mean
DT _{sib}	0.32	0.26	0.26	0.28	0.29	0.28
DT _{anc}	0.37	0.22	0.22	0.27	0.28	0.27
DT _{pq}	0.35	0.28	0.26	0.29	0.28	0.29
DT _{inv}	0.38	0.32	0.27	0.26	0.32	0.31

(b) Train with non-English, test with English documents.

Table 4.5.: Influence of the DT-gram shape σ on the classification performance for the tested datasets measured in macro F_1 .

ety of possible shapes. In this thesis, a preliminary round of evaluation experiments was used to narrow down a large list of candidate structures which was modeled after the work of Pobitzer [71], which is included in Appendix A. The four most promising candidates which were selected are displayed in Figure 4.1. Summarized, they include two shapes that capture simple structures (DT_{anc} for ancestry and DT_{sib} for sibling relationships) and two more complex shapes that combine them (DT_{pq} and DT_{inv}).

Table 4.5 shows the performance of the tested shapes across the different Reddit datasets, for each of the classification approaches tested. Table 4.5a shows the macro F_1 score of the model that received the English documents as training data and predicted documents of the respective other language. It can be seen that the influence of the DT-gram shape σ is not equal in all datasets. For the R4-NL dataset, the choice of σ hardly matters (0.26 vs. 0.27), while it is much more significant for the R1-DE dataset (0.34 vs. 0.40). This effect is also visible in Table 4.5b, where the macro F_1 scores of the opposite evaluation direction are displayed: the model is trained using the non-English documents and predicts the authorship of the English documents.

The results further show that the shapes covering only one “dimension” of the dependency graphs generally perform lower than the shapes that have two size dimensions. This suggests that more complex syntactic features are more expressive in terms of authorship classification for all languages analyzed in this experiment. Concretely, the shape DT_{pq} shows the best perfor-

mance when training with the English documents, while DT_{inv} is superior when training with the non-English documents.

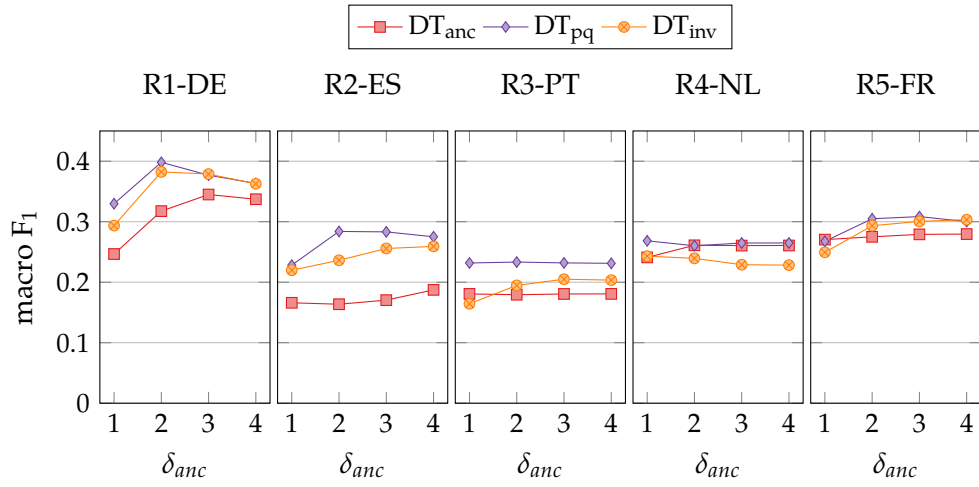
Another interesting result is the performance difference across the different datasets. The German-English dataset R1-DE shows the highest F_1 score in both classification directions (0.40 for training with English documents, 0.38 for the inverse direction). While objective comparisons between language complexities are difficult, research suggests that German can be seen as grammatically more complex than the other language pairs analyzed in this work [76]. Hence, together with the higher scores of the larger DT-gram shapes, one explanation for the observed behavior is that using DT-grams for cross-language classification is more effective on grammatically complex languages. However, more experiments including different language combinations must be performed to strengthen this claim.

4.6.4. Influence of DT-gram Dimensions

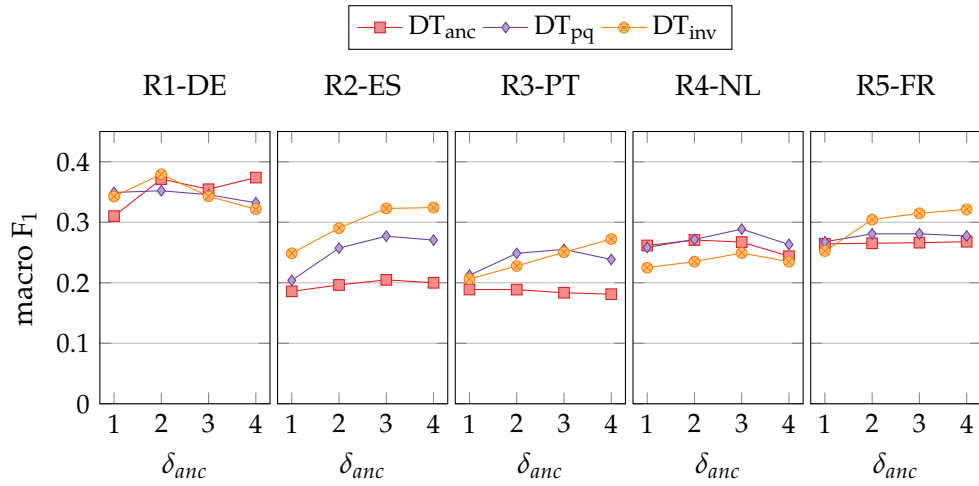
Figures 4.6 and 4.7 show how the choice of the DT-gram dimensions is affecting the classification accuracy. Note that not all shapes appear in both figures as not all shapes have both parameters (specifically, DT_{sib} does not have a δ_{anc} parameter and not appear in Figure 4.6, and likewise, DT_{sib} is not in Figure 4.7). From these results, several interesting conclusions can be drawn:

- For some language combinations, the size of the dimensions hardly has an influence on the best classification score. For example, changing δ_{anc} has little impact on the F_1 scores for the Dutch dataset. In these cases, selecting a small value for the dimension is preferable, as it reduces the size of the feature space (cf. Table 4.2) and therefore the computation time required.
- The selection of training direction has a major impact on the classification behavior. For example, increasing δ_{anc} in DT_{pq} increases the accuracy on the French dataset when training with English and testing with French documents (third graph of Figure 4.7a), but decreases when evaluating the other way around (third graph of Figure 4.7b).
- For most shapes, the best value for δ_{anc} is between 2 and 3, which places them in the same magnitude area as widely used values for character and word n -grams.
- The most suitable value for δ_{sib} for most shapes is between 1 and 2, although the DT_{sib} shape seems to profit from larger values.

4.6. Finding Optimal DT-gram Parameters



(a) Train with English, test with non-English documents.



(b) Train with non-English, test with English documents.

Figure 4.6.: Influence of the size of dimension δ_{anc} (number of ancestors) on the classification results. Each value represents the highest macro F_1 score reached over all classifiers and node representations.

All in all, the two more complex shapes DT_{pq} and DT_{inv} show the best performances in the experiments, with $\delta_{anc} \in [2, 3]$ and $\delta_{sib} \in [1, 2]$.

4.6.5. Influence of DT-gram Node Representations

As described in Section 4.4, the DT-grams allow three ways to represent a node within the dependency graph when performing cross-language classi-

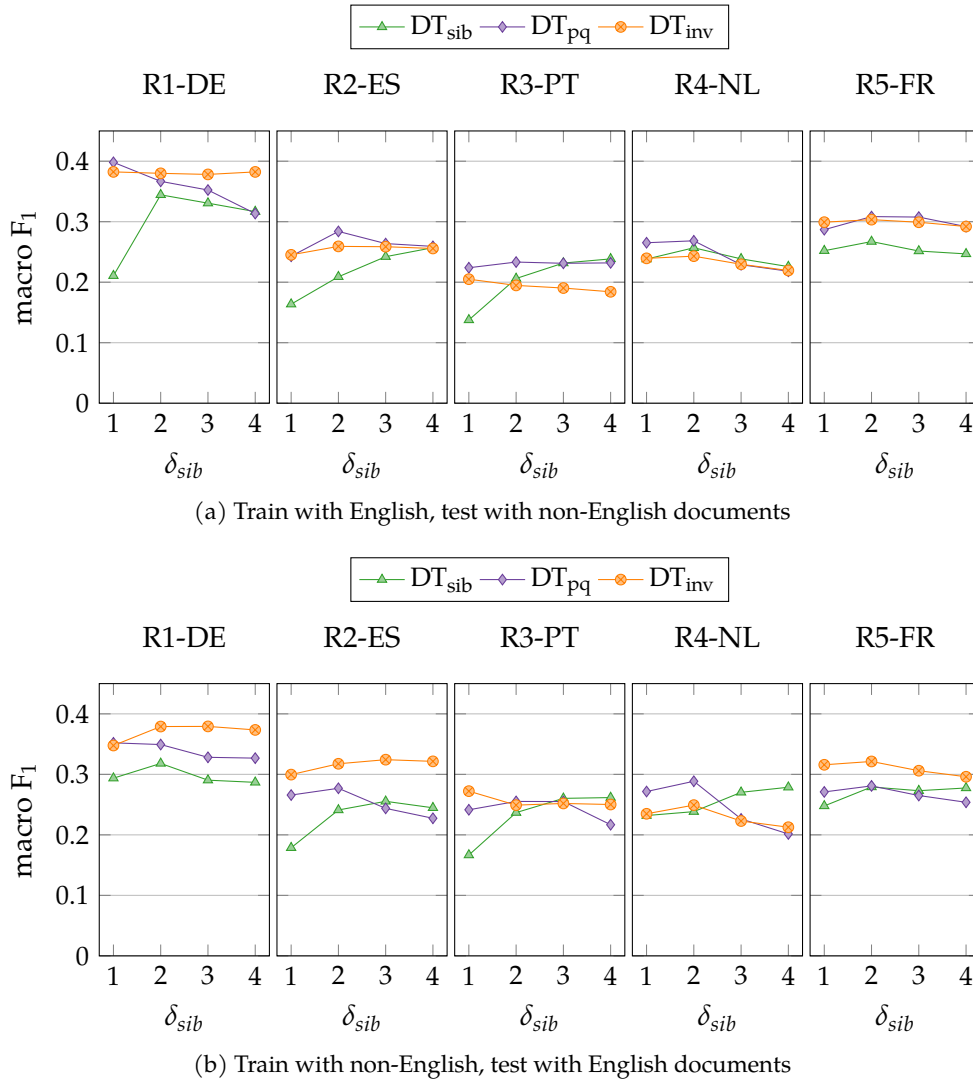


Figure 4.7.: Influence of the size of dimension δ_{sib} (number of siblings) on the classification results. Each value represents the highest accuracy reached over all classifiers and node representations.

classification tasks: by the dependency role of a word to its parent node, by its universal POS tag, or by a concatenation of both. In Table 4.6, the results of these choices is displayed. The results show that using the universal POS tag as node representation (denoted as “pos” in the table) outperforms the other node representations for almost all datasets, with the R1-DE dataset being the only exception (although the results are very close).

Similar to the influence of the shape σ , the role of the node representation η is depending on both the dataset used, as well as the direction in which train-

ing and testing data are chosen from the dataset. Compared to the DT-gram shape experiments, the dataset has less effect, especially when training with the English documents (Table 4.6a). In these cases, the differences between the different values for η hardly exceed 0.02 macro F_1 , and only the R2-ES dataset shows a meaningful difference between the performance of the candidate values.

Table 4.6b shows the other direction of evaluation, where the model is trained with the non-English documents. Here, the influence of η is clearly higher than in the reverse evaluation direction. Similar to the experiments analyzing σ , this observation is strongest for the R1-DE dataset.

At this point, we speculate on possible reasons for this behavior, which is difficult to conclude findings from without having additional metadata surrounding the dataset. For example, the different performances across the two evaluation directions imply that authors may be able to utilize more grammatical versatility in one language than the other, depending on which language is their first language. However, without this information, the interpretation of the results remains difficult.

Note that this experiment measures the effect of the inclusion of the *label* of the dependencies (e.g., “nsubj”). The tree structure from which the DT-grams are extracted is still produced using the dependency grammar parsing, irre-

η	R1-DE	R2-ES	R3-PT	R4-NL	R5-FR
dep	0.40	0.25	0.23	0.26	0.29
pos	0.38	0.28	0.24	0.27	0.31
pos + dep	0.39	0.22	0.23	0.27	0.30

(a) Train with English, test with non-English documents

η	R1-DE	R2-ES	R3-PT	R4-NL	R5-FR
dep	0.34	0.25	0.25	0.28	0.27
pos	0.38	0.32	0.27	0.29	0.32
pos + dep	0.37	0.27	0.26	0.27	0.30

(b) Train with non-English, test with English documents

Table 4.6.: Influence of the word representation η within the dependency graph. Each value represents the highest accuracy reached over all classifiers and DT-gram shapes. “dep” denotes the universal dependency role name, whereas “pos” denotes the universal POS tag (cf. Section 4.4 and Figure 4.3).

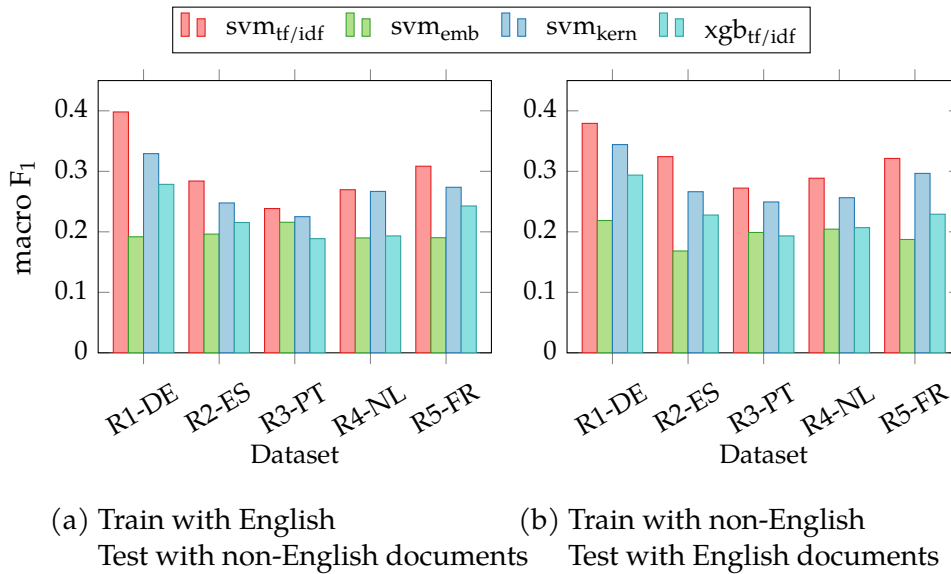


Figure 4.8.: Performance of different methods using DT-grams as feature in the machine learning process.

spective of the node representation. Later in Chapter 5, experiments show that using the dependency grammar as a measure of word distance instead of the original word order does increase the classification performance.

For scenarios where the classification is not *cross-language*, different node representations can be used as well, including a word’s lemma or language-specific POS tag. This strategy is further explored in combination with machine translation in Chapters 5 and 7.

4.6.6. Performance of Models and DT-gram Representations

We use four different classification models in the experiments: $\text{svm}_{\text{tf/idf}}$, $\text{xgb}_{\text{tf/idf}}$, svm_{emb} and svm_{kern} (cf. Section 4.6.2). Figure 4.8 displays the performance of each of these models for the datasets tested. For all datasets, using the SVM in combination with *tf/idf*-normalized frequencies of DT-grams yields the best performance, followed by the kernel methods.

The approaches using the kernel methods and embeddings have several parameters that have been explored (cf. Table 4.4). Figure 4.9 shows the results of the SVM classifier using the kernel methods discussed in Section 4.5.3. The presence kernel, which ignores the number of co-occurrences of features and only counts how many features two documents have in common at all, shows

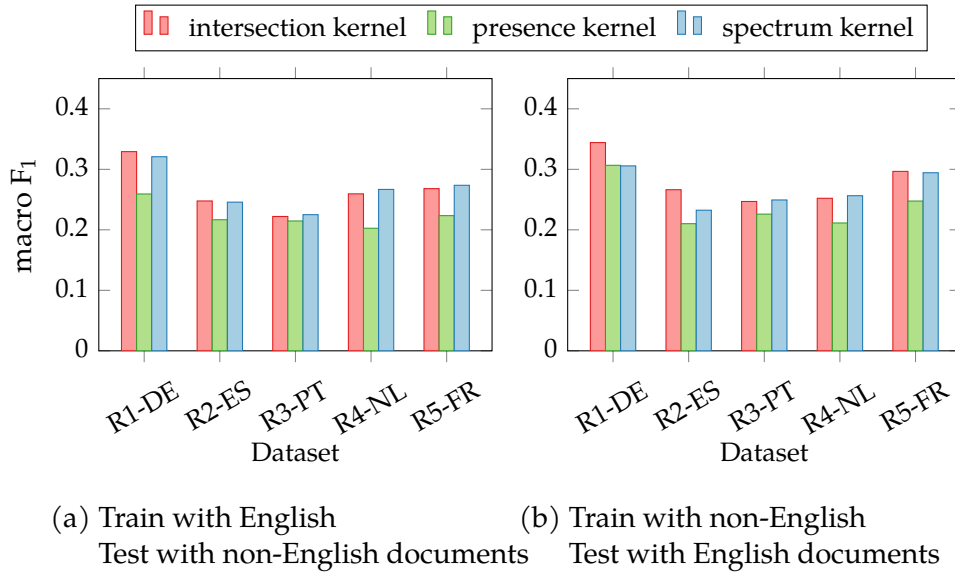


Figure 4.9.: Performance of different kernel methods used by a SVM using DT-grams as tokens.

the lowest results. This confirms the intuition that the additional frequency information of how often authors use certain terms is valuable for authorship attribution.

The embeddings show the worst performance and are not capable of effectively representing the writing style of the authors in the datasets. Figures 4.10 and 4.11 show the effect of the embedding size and algorithm on this result respectively and demonstrate that the approach is generally not performing well for the task at hand. Both parameters have little influence on the F_1 score, which is far behind the other tested approaches.

4.7. Conclusion

In this chapter, a family of versatile text classification features named DT-grams is presented. It is based on a combination of dependency grammar parsing and universal POS tags and allows controlling four hyperparameters: substructure shape (σ), word representation (η), and two dimensions of the shape (δ_{anc} and δ_{sib}).

Additionally, experiments analyzing the influence of the parameters of the DT-gram feature family were presented. The most promising parameter com-

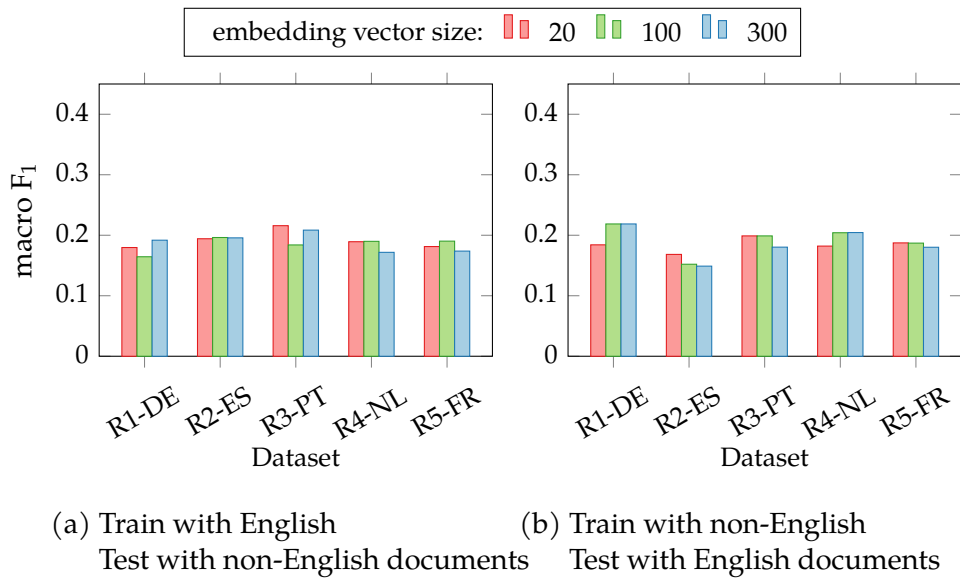


Figure 4.10.: Influence of the embedding vector size on the classification performance.

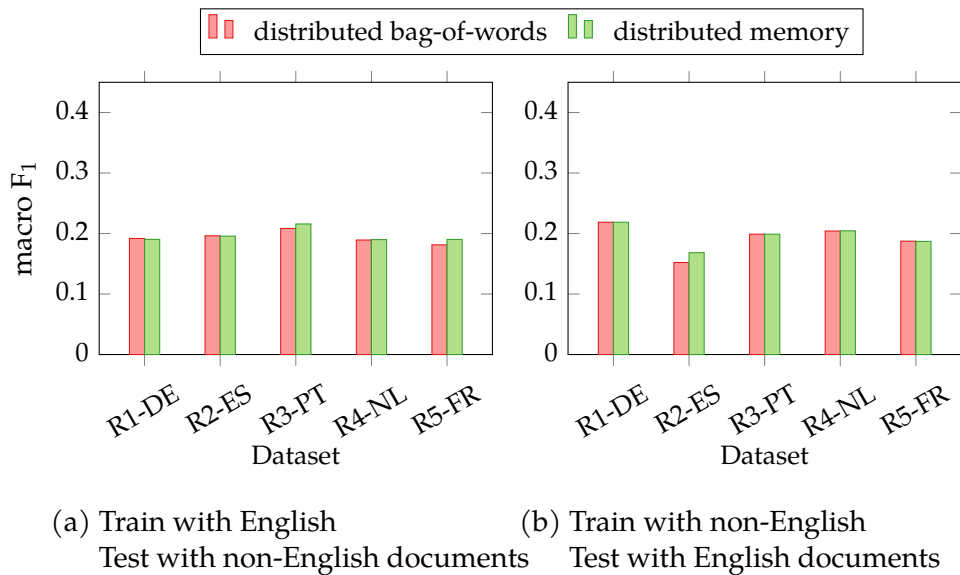


Figure 4.11.: Influence of the embedding vector size on the classification performance.

bination is constituted of using complex dependency graph substructures (DT_{pq} and DT_{inv}), in combination with dimension sizes of 1-2 (δ_{anc}) and 2-3 (δ_{sib}), respectively. While the absolute classification scores of the tested approaches vary across the datasets, the influence of these parameter settings is surprisingly universal.

Furthermore, the experiments show that using universal POS tags as representation for the nodes of the dependency graph yields the highest classification scores and that using a linear support vector machine outperforms the other models tested in the experiments.

Concretely, Table 4.7 shows that the highest average score is achieved by using the DT_{pq} shape with $\delta_{anc} = 2$ and $\delta_{sib} = 3$, which are the exact values that both Augsten [3] and Tschuggnall [94] use in their work. In the remainder of this thesis, tf/idf normalized frequencies of DT-grams using this parameter combination is used in combination with the linear support vector machine as classification model.

classifier	η	δ_{anc}	δ_{sib}	σ	macro F_1
svm _{tf/idf}	pos	2	3	DT_{pq}	0.280
svm _{tf/idf}	pos	2	2	DT_{pq}	0.273
svm _{tf/idf}	pos	2	4	DT_{pq}	0.272
svm _{tf/idf}	pos	4	2	DT_{inv}	0.264
svm _{tf/idf}	pos	4	1	DT_{inv}	0.263
svm _{tf/idf}	pos	3	2	DT_{inv}	0.262
svm _{tf/idf}	pos	3	3	DT_{inv}	0.262
svm _{tf/idf}	pos	4	3	DT_{inv}	0.261
svm _{tf/idf}	pos	3	4	DT_{inv}	0.261
svm _{tf/idf}	pos	3	3	DT_{pq}	0.258

Table 4.7.: The ten combinations of the DT-gram parameters *word representation* (η), *shape dimensions* (δ_{anc} and δ_{sib}) and *shape* (σ) with the highest average macro F_1 score across all datasets.

Evaluating DT-grams on Cross-Language Authorship Attribution

In the previous chapter, the focus of the evaluation experiments was to provide the best possible parameter combinations for the different language pairs that are available in the Reddit datasets presented in Chapter 3. While this provides a solid baseline and helps to choose a model configuration based on the problems at hand, it does not show how well the DT-grams feature performs in comparison to other existing methods that are available for cross-language authorship attribution. In this section, that gap is addressed by comparing the results of solutions based on the DT-grams and their configuration presented in the previous chapter to several state-of-the-art methods used in related natural language processing fields. Concretely, we compare three different methods of cross-language text classification in general: (i) using language-independent features on language-agnostic models, (ii) using language-independent end-to-end classification models, and (iii) using machine translation to change the problem to a single-language classification task. The experiments performed in this regard show that for the small-scale social media datasets at hand, the DT-grams model provides a benefit compared to comparable solutions. However, more interestingly, the best results can be obtained by using DT-grams in combination with machine translation technologies, which enables the usage of language-dependent node representations of the dependency graph.

5.1. Introduction and Related Work

A fundamental property of classification tasks is that predicting the outcome class of an unknown document relies on a common set of features between training and testing data. However, many machine learning techniques that

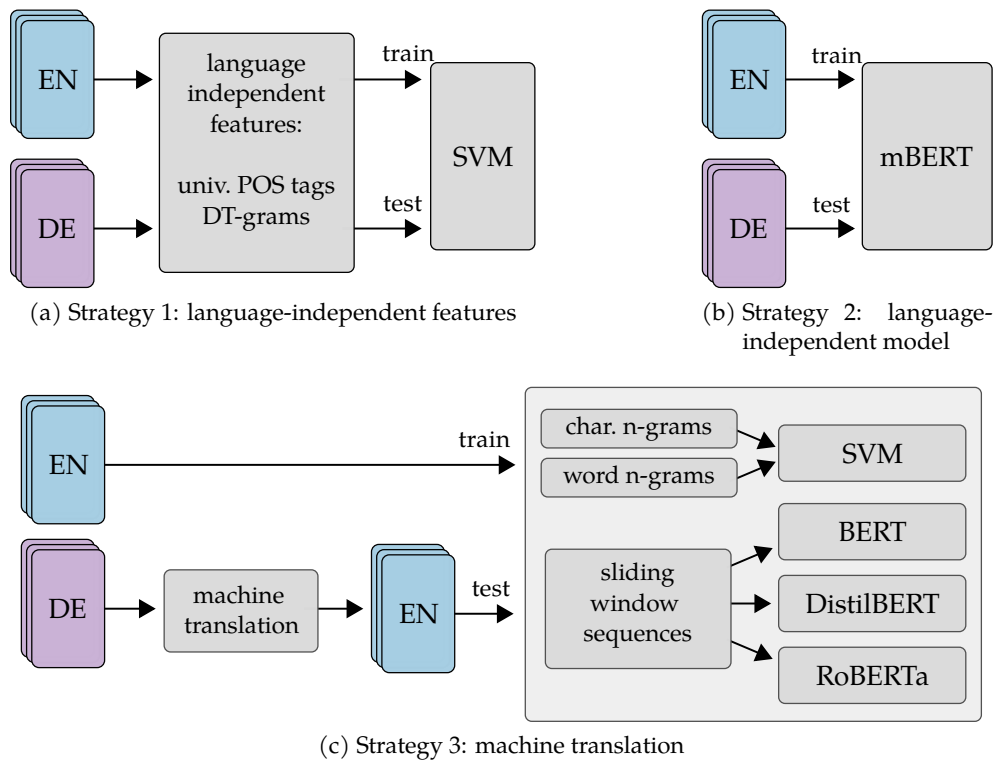


Figure 5.1.: Different strategies used in the cross-language attribution experiments

are successful in single-language tasks are relying on words and characters. If the training and testing documents don't share any words or are even written using different alphabets, it becomes difficult or impossible for the models to measure document similarity. This leads to several fundamentally different strategies on how to solve these tasks, which are also depicted in Figure 5.1:

1. Find features that are not dependent on the language of the texts. For example, by utilizing the DT-grams feature presented in Chapter 4, the language gap can be overcome. However, the fidelity of the feature space is reduced by the mapping of language-specific POS tags and dependency rules to a universal space, which may influence the classification accuracy in a negative way. Other candidates for this category include traditional lexicographic features such as the average sentence length or the type-token ratio, as used by Llorens et al. [52] on a translated cross-language authorship attribution problem.
2. Find machine learning models that are inherently able to classify documents in multiple languages. Pre-trained language models that have

used documents of multiple languages for pre-training are readily available online, and language models have been shown to be effective as cross-language document classification tools [99, 41].

3. Translate the documents of one language into the respective other language using machine translation. This removes the cross-language barrier and enables existing single-language solutions, but introduces a processing step that is both time-consuming and potentially alters the documents in a non-desirable fashion. In the case of *translated* cross-language datasets for authorship attribution (cf. Chapter 3), Bogdanova et al. [5] have shown that this translation step doesn't negatively affect the classification performance, but in these cases, the original author did not write documents in both languages. Hence, in the presented work, one contribution is that the influence of the machine translation step on the classification of true multilingual authorship datasets is analyzed.
4. The results of Bogdanova et al. [5] further suggest that machine translation also aids the performance of language-independent features. We include this possibility by running the experiments with the language-independent features and the multilingual pre-trained BERT model on the machine-translated documents as well.

In the remainder of this section, classification experiments are presented that compare the performance of each of the above methodologies using five of the Reddit datasets from Chapter 3, covering language pairs containing English and one of Arabic, German, Spanish, French and Dutch documents, respectively.

In addition to the four datasets already discussed in the previous section, the English-Arabic dataset R6-AR is added to the collection, while the R3-PT dataset is removed due to technical issues with the machine translation for this language pair.

5.1.1. Experiment Setup

The following paragraphs describe the models and features used in detail, and their parameters can be found in Table 5.1.

5.1.2. Strategy 1: Language Independent Features

We employ two different features for the first strategy: (i) n -grams of universal POS tags, and (ii) DT-grams. The tested configurations for these features are listed in Table 5.1. Note that for the experiments in this section, only four of the DT-gram shapes are considered candidates for the grid search to reduce the parameter search space significantly. Comparing the performance of DT-grams to the universal POS tag n -grams gives insights into whether the structural information that is added by the dependency graph adds to the classification performance of the underlying model.

Originally, an additional feature was tested for this experiment which was based upon the vocabulary richness of documents (modeled after the LIFE features from Llorens et al. [52]). However, the performance of this feature was hardly beating random baselines, and while it shows good performance on novel-length documents, it is not suitable for short social media texts.

5.1.3. Strategy 2: Multilingual Pre-Trained Language Model

The multilingual version of the BERT language model [17] is used as a representative for language-independent models. It was pre-trained using 104 languages and has been proven effective in other cross-language document classification tasks [99, 41]. We use a maximum sequence length of 256 tokens for the BERT model and therefore split the training data into chunks of that size. By making the chunks overlap by 20%, we ensure that all sentences that might be split in half by splitting into chunks are also contained wholly in a different chunk.

5.1.4. Strategy 3: Machine Translation

For the third strategy, we use the open source¹⁵ machine translation library Marian NMT¹⁶ [37] to transform the cross-language into a single-language problem. We do this by translating the non-English documents into English. While the library also contains models for translating the other way around, this direction has the benefit that more single-language models are available for English than for other languages. The now purely English classification

¹⁵We refrain from using commercial products like Google Translate in order to keep research results reproducible.

¹⁶<https://marian-nmt.github.io/>

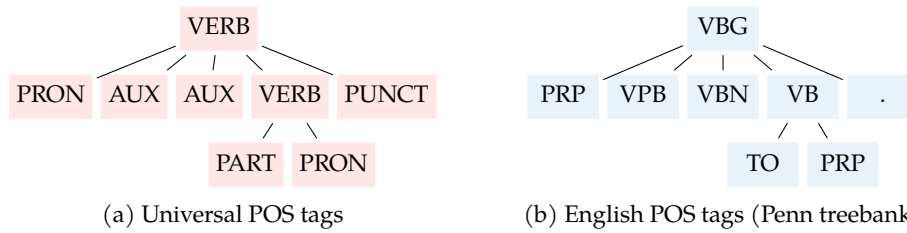


Figure 5.2.: Difference between using universal POS tags (a) and using language-specific POS tags as node representations of the sentence “I have been trying to reach you.”

tasks are then tackled with widely used features and models, including character and word n -grams and several different pre-trained language models.

The original DT-grams as described in Chapter 4 are designed to utilize language-independent features of the dependency parse graph of sentences. By using only one language, however, it is also possible to use language-specific POS tags as node representations inside the graph, as is depicted in Figure 5.2. This changes the DT-grams accordingly and increases the feature space by having a larger token vocabulary. Similarly, instead of containing universal POS tags, using n -grams of language-specific POS tags potentially increases the amount of author-specific information that is captured.

5.1.5. Strategy 4: Combinations

The fourth strategy consists of language-independent approaches that are applied to the machine-translated versions of the datasets. While we don’t expect the performance of these models to exceed the language-specific solutions of strategy 3, we include these experiments to validate the findings of Bogdanova et al. [5], which suggest that machine-translating the documents also significantly improves language-independent approaches.

5.2. Results

Table 5.2 shows the evaluation results of the three presented strategies on five Reddit datasets. Thereby, interesting results for all approaches can be seen:

parameter	value
hyperparameters (used in grid search)	
character, word, univ. POS tag n -gram size	1, 2, 3
σ : DT-gram shape	$DT_{anc}, DT_{sib}, DT_{pq}, DT_{inv}$
$\delta_{anc}, \delta_{sib}$: DT-gram dimensions	1, 2, 3, 4
SVM regularization factor C	0.1, 1, 10
language model parameters (static)	
fine-tuning epochs	3
max. sequence length	256
learning rate	$4 \cdot 10^{-5}$
batch size	8

Table 5.1.: Parameters used for the features and classification models in this section.

Comparing the results of the first approach (universal POS tag n -grams and DT-grams) shows that including the structural information of the dependency graph increases the classification score for all datasets. The extent of the improvement differs between language pairs and is most clearly visible for the German/English dataset, which shows the overall highest scores using this strategy.

Strategy 2 uses the multilingual version of the BERT language model. It performs slightly better than the DT-grams approach for most languages but takes a significant hit for the German/English dataset. This makes it difficult to generalize a comparison between those strategies, as the results suggest that it depends on the language combinations which of the strategies is to be preferred. However, this difference can also be an advantage as the two strategies use mostly disjoint feature sets; while the DT-grams only work on content-independent grammar features, the mBERT model operates on the words themselves. This suggests that the two models are good candidates for ensemble models, which is a possibility that is left as an option for future work.

All of the above models are outperformed by most approaches that use strategy 3, including the machine-translated texts of the datasets. While simple models such as character and word n -grams already achieve consistently higher F_1 scores compared to the previous approaches, the pre-trained language models can't quite keep up with these scores. We suspect that these language models require larger amounts of text to be able to reliably perform any kind of text classification, and further analyze this claim in Section 6.5.

Especially DistilBERT [78], which uses knowledge distillation to produce a reduced version of the original BERT model, shows very low classification scores. More interesting is the performance of the grammar-based features applied to the machine-translated texts. Both the universal POS tag n -grams as well as the DT-grams that use language-specific POS tags as node representations outperform all other approaches in this strategy, except for the Arabic/English dataset.

These results can be compared directly to the findings of the fourth strategy, where the same approaches use the language-independent universal POS tags. It becomes clear that the increased vocabulary is beneficial to the classification process for both the POS tag n -grams as well as the DT-grams features. The mBERT model also profits from the machine translation pre-processing step and thereby confirms the findings of Bogdanova et al. [5] who observe similar behavior for human-translated datasets.

model	R1-DE	R2-ES	R4-NL	R5-FR	R6-AR
<i>approach 1: multilingual features</i>					
univ. POS tag n -grams	0.26	0.29	0.23	0.24	0.11
DT-grams	0.40	0.32	0.26	0.28	0.18
<i>approach 2: multilingual BERT</i>					
multilingual BERT	0.24	0.38	0.25	0.37	0.23
<i>approach 3: machine-translated documents</i>					
character n -grams + SVM	0.41	0.44	0.40	0.44	0.37
word n -grams + SVM	0.36	0.43	0.39	0.41	0.38
BERT	0.27	0.34	0.31	0.42	0.29
DistilBERT	0.16	0.19	0.16	0.19	0.16
RoBERTa	0.26	0.38	0.31	0.43	0.30
lang. spec. POS tag n -grams	0.47	0.46	0.42	0.43	0.34
lang. spec. DT-grams	0.47	0.47	0.43	0.46	0.34
<i>approach 4: combined</i>					
univ. POS tag n -grams	0.32	0.39	0.35	0.36	0.26
DT-grams	0.44	0.46	0.42	0.45	0.33
multilingual BERT	0.27	0.32	0.31	0.42	0.29

Table 5.2.: Evaluation results of all strategies, measured in macro-averaged F_1 score.

5.3. Conclusion

This chapter also shows experiments using DT-grams in various approaches for CLAA, including several widely used text classification baselines. The most important result is that if machine translation is feasible and available, it can be utilized to significantly increase the performance of almost all approaches. For the analyzed language combinations, the DT-grams feature presented in this thesis then outperforms all other approaches when using language-specific POS tags as node representations, followed by simpler grammar-based features and lexicographic features. At least for the short texts used in this section, the pre-trained language models have difficulties in attributing the correct authors, and the knowledge distillation further deteriorates this capability. In scenarios where no machine translation is available, the multilingual BERT, as well as the DT-grams model, show promising results in the final scores, but more experiments using ensemble methods are suggested for these specific models.

Summarized, the evaluations have demonstrated that DT-grams are an efficient feature for CLAA on short social media texts, providing an answer to the second research question: *Which language-independent syntax-based features are a viable choice for a classification feature for CLAA?* - DT-grams.

Evaluating DT-Grams on General Authorship Attribution¹⁷

Until this point in the thesis, features and methods for cross-language text classification have been explained in detail, and the appropriate datasets for the evaluation experiments have been presented. In this chapter, the evaluation strategies for authorship attribution in particular are analyzed from a more general point of view. Thereby, the data bias which is introduced by using not enough different evaluation datasets is focussed upon. This chapter explains how a wide variety of different existing authorship attribution datasets aims to mitigate this bias, and how several aggregated scores of different dataset properties can help to characterize the properties, strengths, and weaknesses of proposed machine learning models and features.

While the previously presented DT-gram features are developed with the specific aim of cross-language authorship attribution, analyzing their performance in various other scenarios allows to better estimate the generalizability of the approach, and this chapter presents the strategy for evaluating DT-grams in a bigger picture.

6.1. Introduction and Related Work

An important aspect of any field of research that uses data to evaluate theories and models is the generalizability of the results found. Research in authorship attribution in particular often does not give much weight to this as-

¹⁷Results and contents of this chapter are based on and partially reused from the paper: Benjamin Murauer and Günther Specht: *Developing a Benchmark for Reducing Data Bias in Authorship Attribution*. In 2nd Workshop on Evaluation & Comparison of NLP Systems (Eval4NLP'2021), pages 179–188, 2021.

pect. Many previous studies use either only one dataset or don't specifically increase the diversity of the datasets used, and additionally, often fail to address this implicit data bias: Even foundational work in this field trying to categorize features in this field in a fundamental way can be prone to this issue. For example, Grieve et al. [28] measure the effectiveness of 39 different feature types for attribution. They address the importance of the dataset being representative of a language and explicitly explain the characteristics of the texts and authors in great detail, but consequently, by using a single dataset, their findings of feature performances are restricted to those very characteristics. Nevertheless, findings of such fundamental work are often used as a reference in research using completely different datasets.

One idea to mitigate any bias in the content of a dataset is to focus on the separation between *style* and *content*. This can be achieved by explicitly modeling the topic [81] or by using cross-topic or cross-domain datasets, where the training data and the test data have a different genre or contain texts about different topics [86, 79, 40]. For the latter, the key idea is that by minimizing the topic or genre-specific content contained in the overlap of training and testing data, any performances measured must conclude from the stylistic information from the authors. For both approaches, however, the bias towards those authors remains in the evaluation, and it remains unclear whether any resulting insights generalize to other authors, or are specific to the authors of the selected dataset.

Even from within a dataset, the choice of training and testing data can have a large impact on the outcome and additionally varies across languages [22]. Additionally, Eder et al. [20] demonstrated that the amount of text required to reliably attribute an author also depends on the language, and suspect that this result may be depending on the genre of text as well. Similarly, Luyck et al. [54] show that while some feature types are more robust to the size of the dataset, the performance of others varies greatly depending on the number of documents per author and the number of authors.

It is therefore difficult to determine any stylistic aspect of writing that is able to determine an author's identity that holds *in general* and does not depend on the document's length, language, topic or other characteristics. To determine such features, evaluations on datasets with all of these aspects have to be made.

In this chapter, the authorship attribution benchmark is presented, containing as many different datasets as possible, with many different aspects that try to cover the broadest possible landscape of authorship attribution problems. While the focus of this thesis lies on cross-language classifications, this benchmark includes a multitude of different scenarios, enabling a quick and

thorough comparison of the performance of any features designed for cross-language application to previous examples, also on single-language setups or different text genres.

This chapter uses the presented benchmark to evaluate DT-Grams in a broader context and showcases its performance in a wider variety of authorship attribution problems. Therefore, we compare its performance to several widely used text classification methods, including SVMs operating on character n -grams and pre-trained language models like BERT [17].

6.2. Dataset Characteristics and Metrics

Comparing datasets requires measurable features that can be obtained objectively and reliably. In this section, the dataset aspects that were considered when compiling the corpus benchmark are explained. Note that this section will use symbols and terminology introduced in Chapter 3.

Summarized, the following metrics are considered for the datasets in this chapter:

i Dataset Metrics

definition	description
$ A_D $	number of authors in the dataset
$ D $	number of documents in the dataset
dl_n_w	lengths (in words) of all documents in the dataset
sl_n_w	lengths (in words) of all sentences in the dataset
dpa	number of documents for each author in the dataset
imb	author imbalance for each author in the dataset (cf. Section 3.9)

6.3. Datasets used in the Benchmark

This section presents the datasets that are used in the authorship attribution benchmark. For each dataset, detailed information on how they are obtained is provided. Additionally, for each dataset, detailed instructions on how the

dataset should be used in evaluation experiments are suggested by explaining the strategies that should be used to extract training and testing samples from the dataset. This is an important step towards ensuring that experiments remain comparable, as the selection of train/test splits plays an important role in the reproducibility of experiments [21].

Table 6.1 shows an overview of the presented datasets along with linguistic statistics as described in Section 6.2. It shows the high variance in many of these aspects: except for the average sentence length σ (sln), every column contains a wide range of values.

6.3.1. CCAT50

The CCAT50 dataset [51] is a subset of the Reuters Corpus vol. 1 [47] and contains news articles of the corporate/industrial category. It is balanced and homogeneous and features a pre-defined split of 50 training and 50 testing documents for each of the 50 authors.

In addition to its original configuration, we leverage the sufficiently large size of the dataset in terms of the number of documents per author and add an additional train/test split strategy where the number of training documents per author is limited. This way, the effect of the number of training documents can be analyzed with arbitrary numbers, rather than being limited to a specific value determined by the dataset itself. This limited version of the dataset will be referred to as CCAT50_{sm}¹⁸.

6.3.2. CL-Novels

The CL-Novels dataset was introduced by Bogdanova et al. [5] when introducing the task of cross-language authorship attribution. It consists of novels by English authors, and Spanish translations of some of the works of those authors. Table 6.2 shows the titles used in the thesis.

The dataset was reconstructed according to the information provided by Bogdanova et al. [5] by obtaining the texts from the Gutenberg project¹⁹. As the original authors, we split the works into chunks of 500 sentences. Thereby,

¹⁸This version is not included in Table 6.1 as the metrics depend on the number of authors and documents chosen

¹⁹<https://gutenberg.org>

dataset	$ A_D $	$ D $	\overline{dln}	$\sigma(dln)$	\overline{sln}	$\sigma(sln)$	\overline{dpa}	$\sigma(dpa)$	\overline{imb}	$\sigma(imb)$
CCAT50	50	5,000	588	158.6	27.0	3.7	100.00	0.0	141.5	35.1
CL Novels	6	298	10,161	3,724.4	20.9	4.9	49.6	19.0	3205.1	616.2
CMCC	21	756	660	492.6	25.2	16.7	36.0	0.0	413.8	242.2
Guardian	13	444	1,226	454.1	23.3	4.5	34.1	7.0	370.7	206.6
IMDb62	62	61,965	350	229.8	26.0	9.4	999.4	1.3	134.0	68.8
PAN18-AA	20	1,496	973	91.1	18.0	7.6	74.8	35.2	90.6	18.8
R1-DE	28	4,087	720	307.3	23.0	10.0	145.9	128.8	269.0	76.3
R2-ES	20	4,450	641	267.2	24.1	8.8	222.5	203.9	233.4	77.7
R3-PT	37	4,481	614	233.7	23.1	10.5	121.1	82.7	227.0	64.7
R4-NL	11	2,410	649	297.2	19.8	5.2	219.0	136.9	265.8	78.2
R5-FR	45	10,131	637	285.4	24.2	8.1	225.1	173.2	256.5	81.4

Table 6.1.: Linguistic statistics of datasets used in the authorship attribution benchmark.

we relied on sentence splitting provided by the *stanza*²⁰ library, which we also used for parsing the sentences. Unfortunately, no further details regarding preprocessing steps are provided in the original publication. We additionally performed the following steps:

- we removed the preamble of each title. This is a text that is added by the Gutenberg project which describes the origins of the work and will always contain the name of the author and the title of the novel. An example of such a preamble can be found in Appendix C.1.
- we removed any appendices. This usually includes the full license under which the work is published by the project Gutenberg, which can be found online²¹, and sometimes also includes unrelated text (e.g., advertisements) that happens to be part of the concrete version of the novel published by the project.

We use the dataset as a cross-language evaluation dataset, so the languages of the documents used for training are disjunct from those of the testing documents (cf. Section 3.1). As is visible in the table, some authors feature the same novel in multiple languages (e.g., for Jane Austen, all novels are available in both languages).

This requires a validation strategy where the same novel is never used for training and testing at the same time. Hence, the original authors of the dataset suggest a *leave-one-novel-out* evaluation strategy, a variation of the traditional leave-one-out strategy [16]. Here, documents of one title (i.e., novel) in the testing language are used for testing, and all documents in the training language that don't have the same title as the testing documents are used for training.

We have adopted this algorithm and optimized it by preventing separate train/test splits that contain the same training documents and only differ from one another in the testing documents, but rather extend the test set. For example, evaluating the Spanish version of the titles *The Ebb-Tide* and *Olalla* by Robert Stevenson will use the same training documents, as both of those titles are only available in Spanish. In this case, instead of training the model twice with the same data, we merely evaluate both titles with the same trained model. The algorithm is displayed in Algorithm 2, with the optimization in lines 7 and 8. Thereby, l_{train} and l_{test} denote the training and testing languages, respectively.

²⁰<https://github.com/stanfordnlp/stanza>

²¹<https://gutenberg.org/policy/permission.html>

author	English titles	Spanish titles
Charles Brontë	vilette (21), the professor (6), jane eyre (20)	jane eyre (23)
Jane Austen	pride and prejudice (12), emma (12), lady susan (2)	pride and prejudice (13), emma (17), lady susan (3)
Lewis Carroll	sylvie and bruno (7), the hunting of the snark (1), alice in wonderland (2)	through the looking glass (5), alice in wonderland (4)
Oscar Wilde	the picture of dorian gray (13), the soul of a man under socialism (2), lady windermeres fan (7)	lord arthur saviles crime (2), the picture of dorian gray (13), the happy prince (1)
Robert Stevenson	treasure island (7), the black arrow (10), the strange case of dr jekyll and mr hyde (3), new arabian nights (9)	the ebb-tide (7), treasure island (9), olalla (2)
Rudyard Kipling	captains courageous (8), the phantom rickshaw (1), the jungle book (6), kim (18), from sea to sea (22)	the phantom rickshaw (1), the jungle book (9)

Table 6.2.: Authors and novels in the CL Novels dataset. Numbers in parenthesis indicate the number of 500-sentence-chunks that were extracted from the respective novel.

Note that the underlying dataset changes if the training and testing languages are swapped, as only one Spanish title from Charles Brontë is available, and the split where the English version of *Jane Eyre* is used for evaluation, no training data is available for that author. This problem does not occur in the original authors' work, as they only train on English documents and test on the Spanish versions.

6.3.3. CMCC

The CMCC²² dataset was developed by Goldstein et al. [26] by instructing 21 students to express their opinion on 6 different topics in 6 different genres, which are displayed in Table 6.3. The dataset is perfectly balanced in the sense

²²The origin of the widely-used acronym for this dataset remains unknown.

Algorithm 2 The optimized leave-one-novel-out evaluation strategy.

```

1:  $S \leftarrow \emptyset$ 
2:  $D_{train} \leftarrow \{d \mid d \in D \wedge l_d = l_{train}\}$ 
3:  $D_{test} \leftarrow \{d \mid d \in D \wedge l_d = l_{test}\}$ 
4: for  $t \in \{d.title \mid d \in D_{test}\}$  do
5:    $S_{train} \leftarrow \{d \mid d \in D_{train} \wedge d.title \neq t\}$ 
6:    $S_{test} \leftarrow \{d \mid d \in D_{test} \wedge d.title = t\}$ 
7:   if  $\exists (S_x, S_y) \in S$  where  $S_x = S_{train}$  then
8:      $S_y := S_y + S_{test}$ 
9:   else
10:     $S := S + (S_{train}, S_{test})$ 
11:   end if
12: end for
13: return  $S$  ▷  $S$  contains tuples of (train, test) documents

```

genre	topic
interview transcript	church
discussion transcript	gay marriage
chat	war in Iraq
essay	legalization of marihuana
email	privacy rights
blog post	gender discrimination
(a) Text genres in the CMCC dataset	(b) Topics in the CMCC dataset

Table 6.3.: Topics and text genres of the CMCC dataset.

that every student has contributed exactly one document in every text genre for every topic.

This also makes it able to evaluate both cross-topic as well as cross-genre classification setups, depending on the way the data is divided into train/test splits. In this thesis, the CMCC dataset is used for both cross-topic and cross-genre evaluations, and depending on the scenario, the dataset will be referred to as $CMCC_{\times T}$ or $CMCC_{\times G}$.

All evaluations are performed using the *leave-one-topic-out* and *leave-one-genre-out* strategies, respectively, where all documents of one topic (or genre) are used for testing, and all remaining documents are used for training.

author	book review	opinion article			
		politics	society	UK	world
Catherine Bennett	10	10	4	10	10
George Monbiot	0	6	3	3	10
Hugo Young	3	8	6	5	10
Jonathan Freedland	2	9	1	10	10
Martin Kettle	2	7	0	3	10
Mary Riddell	4	8	10	10	10
Nick Cohen	5	10	2	7	9
Peter Preston	10	10	1	10	10
Polly Toynbee	4	10	10	5	10
Roy Hattersley	10	10	4	10	3
Simon Hoggart	2	10	5	6	5
Will Hutton	7	10	6	5	10
Zoe Williams	4	4	10	6	10

Table 6.4.: Distribution of documents by topic/genre in the Guardian dataset

6.3.4. Guardian

The Guardian dataset was developed by Stamatatos [86] and consists of opinion articles and book reviews by journalists of the *Guardian* newspaper. Therefore, along the lines of the CMCC dataset, it can provide evaluation data for both cross-genre as well as cross-topic scenarios, using the previously described *leave-one-topic-out* and *leave-one-genre-out* strategies.

However, unlike the CMCC dataset, it is not balanced. Table 6.4 shows the number of documents provided by each author for each topic and genre. It can be seen that some authors don't provide any documents for a specific topic, making it difficult to compare separate train/test splits directly. For example, when evaluating using documents of the genre *opinion article* for testing, no documents for the author *George Monbiot* are available for training the model.

Analogously to the CMCC dataset, the cross-topic and cross-genre variants of the Guardian dataset will be referred to as $\text{Guardian}_{\times T}$ and $\text{Guardian}_{\times G}$, respectively.

language	problem	authors	documents	$\overline{\#(d)}$
English	1	20	245	4,346
	2	5	56	4,325
French	3	20	189	4,497
	4	5	56	4,526
Italian	5	20	220	4,745
	6	5	81	4,801
Polish	7	20	243	5,169
	8	5	50	5,099
Spanish	9	20	257	4,792
	10	5	99	4,873

Table 6.5.: Sub-Problems of the PAN18-Fanfiction dataset

6.3.5. IMDb62

The IMDb62 dataset contains reviews from the Internet Movie Database ²³ and was compiled by Seroussi et al. [82]. It is part of a larger dataset called the *Prolific IMDb Users* dataset by the same authors and includes the 62 most “active” authors of the platform in terms of written reviews. For each of the authors, 1,000 documents are available. Compared to the other datasets, this is a very high number.

The dataset is not purely constructed for authorship analysis, but also contains the rating information regarding the movies that are reviewed. Therefore, it is also used in other fields such as sentiment analysis.

The IMDb62 dataset is homogeneous and does not feature different topics or genres, and thus, we use a stratified 5-fold cross-validation strategy in the experiments. Like the CCAT50 dataset, its size allows to sub-sample smaller versions of the dataset to obtain additional experiment results. These smaller versions are referred to as IMDb62_{sm}.

6.3.6. PAN18-Fanfiction

The PAN18-Fanfiction dataset contains prose written by fans of specific fandoms. A fandom is a coherent universe that was introduced by a movie, TV show, novel, etc., and often, amateur writers continue storylines of those universes on their own, and these works are referred to as *fanfiction*. The dataset

²³<https://imdb.com>

was compiled by Kestemont et al. [40] and was used to evaluate the shared task of authorship identification in the PAN workshop in 2018²⁴.

It contains documents by authors that write documents for more than one fandom, making it a cross-*fandom* dataset. The entire dataset is divided into 10 sub-tasks, two for each of the languages English, French, Italian, Polish and Spanish. It must be noted that while this dataset is therefore *multilingual*, it is not *cross-lingual*, as each author only writes in one language (cf. Section 3.1).

The dataset features a pre-defined split of training and testing data. Thereby, the training documents are from various different fandoms, whereas the testing documents all belong to the same fandom. This is comparable to the evaluation strategies used for the other datasets presented in this section: in the cross-topic evaluation of the CMCC and Guardian datasets, one topic is used for testing while *all others* are used for training.

6.3.7. Reddit Datasets

In Chapter 3, the framework for compiling datasets of cross-lingual authors is presented, and we use several of these datasets for the evaluations in this chapter.

For the evaluation splits, two different approaches are used.

1. The unaltered version of the datasets as presented in Chapter 3.
For this scenario, each dataset provides exactly two splits; one split returns all English documents as training documents and all documents of the respective other language as testing documents, and the other split reverses this order.
2. A balanced version. Similar to the experiments determining optimal parameters for the DT-grams (cf. Section 4.6), reduced versions of the datasets are used to enable direct comparisons between the cross-language datasets, which are otherwise unbalanced.

Therefore the train/test splits are constructed using Algorithm 1: for each dataset, ten random authors are sampled, and for each of those authors ten random documents are selected. The testing data is left un-

²⁴<https://pan.webis.de/clef18/pan18-web/authorship-attribution.html>

altered. The entire process is repeated five times, each time randomly selecting authors and documents ($p_{it} = 5$ in Algorithm 1).

We are aware that this means that for smaller datasets, the repetitions are more likely to include overlaps in the authors and documents compared to larger datasets, but this limitation is difficult to circumvent without the availability of larger datasets.

The reduced datasets will be referred to as $\text{Reddit}_{\text{sm}}$ in the remainder of this chapter, and will also serve as a dataset with limited training data for the aggregated score calculations later in Section 6.5.2.

6.4. Aggregated Scores

Ultimately, we envision that evaluating novel approaches in authorship attribution and other fields should not be limited to comparing the scores of separate and independent datasets. Instead, the focus should be shifted towards gaining insights into which properties and aspects of the underlying datasets contribute to the observed performance. Answering questions like “How well does feature xy perform on short texts in comparison to long ones, which I have already tested?” should not rely on obtaining a single evaluation dataset that happens to contain shorter documents, but might be different in many other aspects as well. For this reason, we have developed a set of aggregated scores with the goal of showcasing specific model and feature performances for a list of aspects of datasets. In combination with the datasets described previously in this chapter, more robust statements regarding the characteristics of features and models can be made.

We have identified the following scores that aim to combine the performances of a model on datasets that share a specific aspect:

mono describes the performance on datasets that contain one coherent topic, genre, and language. The aim of this score is to measure how well a model is able to distinguish properties of the text that can be closely attributed to the author, as the other factors are ruled out as well as possible.

small measures the model’s capability to perform classification on smaller numbers of training documents for each author. For this measure, different datasets can be used that contain sufficiently many training documents in general, by simply subsampling the number of training docu-

score	datasets included
mono	IMDb62, CCAT50
small	IMDb62 _{sm} , CCAT50 _{sm} , Reddit* _{sm}
mixed-language	PAN18-FF
cross-topic	CMCC _{×T} , Guardian _{×T}
cross-genre	CMCC _{×G} , Guardian _{×G}
cross-language	Reddit*, CL-Novels

Table 6.6.: Constellation of datasets used in the different scores. Reddit* denotes the collection of the Reddit datasets R1-DE, R2-ES, R3-PT, R4-NL, and R5-FR.

ments for each author. In this thesis, we use all datasets that feature more than 50 training documents per author in order to run evaluations with different amounts of training documents. The subsampled datasets are denoted with a suffix _{sm}.

mixed-language is used to determine a model’s capability to work on documents with different languages. However, it is important to note that this does not mean *cross-language*, but merely that the model is able to be trained with different languages effectively (a detailed definition of these terms can be found in Section 3.1). For example, a model using character n -grams is likely to work with different languages, given the testing documents are written in the same language as the training documents. This does not necessarily hold for cross-language setups.

cross-topic/genre describes how well a model performs when the training and testing documents contain text about different topics or are of different text genres. This has been used in the past as a measure to prevent the topical bias of a dataset to influence the model’s ability to attribute the documents.

cross-language shows a model’s ability to work with training and testing documents that have different languages. This is arguably the score that is most difficult to expand in terms of adding datasets.

The distribution of which datasets are used in which scores is listed in Table 6.6.

Note that the aggregated scores can be measured once all experiments have finished once, and no duplicate runs for metrics using the same datasets are

required (other than the repetitions required by datasets using some k -fold cross-validation scheme).

6.5. Evaluating DT-grams

After having shown the efficiency of the DT-grams feature for authorship attribution in a cross-language setup, the next step is to compare the performance in related fields. In this section, the datasets presented in this chapter are evaluated using DT-grams as well as several other classification models widely used in this field.

6.5.1. Experiment Setup and Baseline Models

The datasets involved in the following experiments are described in detail in Chapter 6. For each dataset, a well-defined strategy yielding train and test splits is defined. Table 6.7 summarizes the datasets and shows how many splits each strategy contains. For the experiments, the performance of various models (which are explained shortly) is compared for each of these datasets, whereby the final score for each dataset is calculated by averaging the score of all splits belonging to that dataset. For example, the PAN18-FF dataset consists of 10 pre-defined sub-tasks, so the score for this dataset is calculated by using the mean score of these 10 sub-tasks. Likewise, results of the dataset denoted “Reddit” in the experiments of this chapter denote the average results of the datasets R1-DE, R2-ES, R3-PT, R4-NL, and R5-FR. This aggregation strategy is used to obtain a single evaluation result for the entire dataset, and mitigate bias towards datasets consisting of multiple sub-tasks (or splits) with respect to solitary datasets.

Aside from DT-grams, the authorship attribution benchmark is also tested with the following features and models: A linear SVM paired with tf/idf-normalized frequencies of character 3-grams, word 1-grams, and DT-grams, as well as Doc2Vec embeddings using character and universal POS tag 3-grams, and three pre-trained language models BERT [17], DistilBERT [78] and RoBERTa [50]. A schematic overview of the models is provided in Figure 6.1. Along the lines of the experiments on the Reddit dataset in the previous section, the documents in the datasets in this evaluation are split into chunks of 256 words in order to fit them into the maximum sequence length of the pre-trained language models.

dataset	splits	description
CCAT50	2	predefined (50% / 50%)
CL-Novels	15	leave-one-novel out
CMCC _{×G}	6	leave-one-genre-out
CMCC _{×T}	6	leave-one-topic-out
Guardian _{×G}	2	leave-one-genre-out
Guardian _{×T}	4	leave-one-topic-out
IMDb62	5	stratified 5-fold
PAN18-FF	10	predefined (10 sub-tasks)
Reddit*	10	leave-one-language-out, see Chapter 3

Table 6.7.: Train/test splits for each dataset. Reddit* denotes the collection of the Reddit datasets R1-DE, R2-ES, R3-PT, R4-NL, and R5-FR.

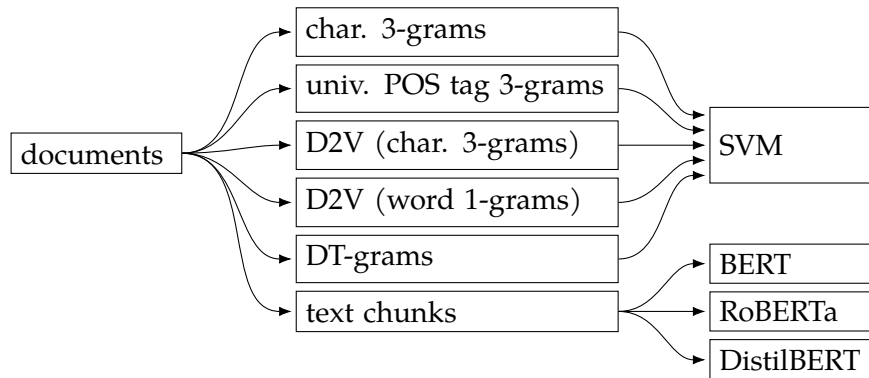


Figure 6.1.: Features and models used in the evaluation experiments.

6.5.2. Results

Table 6.8 shows the scores of the respective models for each dataset in the authorship attribution benchmark. Some unexpected results arise from these raw results. For example, while the DT-grams feature is the strongest approach for the Reddit datasets, the RoBERTa model shows a higher performance for both the larger Reddit dataset and the CL-Novels datasets, which both contain significantly more training data for the respective tasks. This suggests that DT-grams are a strong candidate for low-resource scenarios, but are quickly outperformed by pre-trained language models once sufficient training data is present, confirming the results of similar experiments using single-language datasets (cf. Figure 6.2).

However, the focus of the authorship attribution benchmark is not to rank specific models according to their overall performance, but rather to showcase

dataset	char. 3-grams	univ. POS 3-grams	DT-grams	Doc2Vec (char. 3-grams)	Doc2Vec (word 1-grams)	BERT	DistilBERT	RoBERTa
CCAT50 _{sm}	0.64	0.52	0.44	0.25	0.33	0.59	0.57	0.60
CCAT50	0.70	0.61	0.53	0.30	0.41	0.66	0.67	0.66
CL-Novels	0.18	0.19	0.18	0.09	0.14	0.12	0.11	0.16
CMCC _{×T}	0.61	0.54	0.39	0.15	0.25	0.54	0.48	0.60
CMCC _{×G}	0.69	0.45	0.29	0.21	0.28	0.32	0.29	0.29
Guardian _{×T}	0.82	0.81	0.65	0.41	0.52	0.85	0.81	0.84
Guardian _{×G}	0.53	0.55	0.30	0.29	0.39	0.44	0.44	0.45
IMDb62 _{sm}	0.75	0.66	0.48	0.22	0.26	0.55	0.53	0.51
IMDb62	0.97	0.92	0.87	0.19	0.56	0.98	0.98	0.98
PAN18	0.49	0.38	0.28	0.38	0.22	0.33	0.39	0.42
Reddit _{sm}	0.03	0.15	0.18	0.02	0.01	0.07	0.07	0.14
Reddit	0.05	0.13	0.17	0.03	0.04	0.19	0.13	0.27

Table 6.8.: Results of evaluating DT-grams on the authorship attribution benchmark.

the strengths and weaknesses of specific solutions regarding the properties of the dataset. To emphasize this intentions, the results are analyzed in more detail for the aggregated scores *small*, *cross-language*, and *cross-topic/genre*.

Sensitivity to Document Size

Figure 6.2 shows the performance measured in macro-averaged F_1 score for the models on variations of the IMDb62 and CCAT50 datasets that use a limited number of documents for each author for training the models.

It is clear that while all models expectedly show a better performance with more training data, the extent to which the score changes differs dramatically for the IMDb62_{sm} dataset, where the pre-trained language models are affected much stronger by the smaller numbers of training data, and only catch up to simpler models at 50 training documents per author. Although the two datasets shown in the figure contain texts of similar length (cf. Ta-

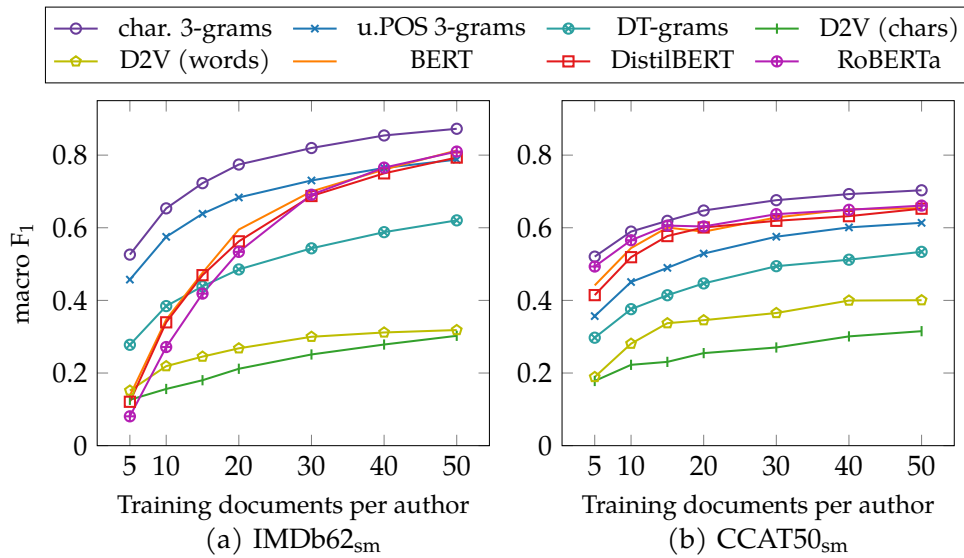


Figure 6.2.: F_1 score of IMDb62_{sm} and CCAT50_{sm} datasets with limited numbers of training documents per author.

ble 6.1 in Chapter 6) this behavior is not visible for the CCAT50_{sm} dataset, where the pre-trained models are able to correctly classify the authors also with few training samples. This indicates that important characteristics like the minimal number of suggested training samples are highly dependent on the dataset, in this case on the topic and text genre, and are difficult to be determined in general (other than trivial recommendations like “more is better”).

The figure also shows that simpler features like character n -grams show significantly superior performances. Interestingly, also the universal POS tag n -grams outperform the DT-grams by quite a noticeable margin, for both datasets analyzed in this scenario. In the experiments shown in the previous chapter, these results are reversed: the DT-grams outperform the universal POS tag n -grams in both the untranslated setting as well as with the machine-learning pre-processing step. This indicates that the advantage that DT-grams have may be dependent on the type of text, and DT-grams are more efficient in classifying social media comments compared to news articles or movie reviews.

Sensitivity to Language

A surprising overall result for the cross-language datasets in Table 6.8 is the relatively high efficiency of the pre-trained language models for the Reddit dataset, as they have not been pre-trained using multilingual texts. This performance is not displayed in the other cross-language dataset containing 19th-century novels, which suggests that his behavior could stem from the genre of texts (social media comments), which are more likely to contain words common in multiple languages than documents from the 19th century. However, we suggest that even more datasets are required to answer this specific question.

Cross-language classification problems are defined by two different choices regarding the candidate languages: Firstly, which languages are considered in the classification problem at all, and secondly, which of those languages are used for training and which are used for testing. Figure 6.3 shows the macro F_1 score of two cross-language models (univ. POS tag 3-grams and DT-grams) and the pre-trained language models for the Reddit dataset. The different colors represent the different language pairs of the Reddit dataset, and the two differently shaded columns of each color represent the classification score (in macro F_1) of both train/test directions used for the experiment (thereby, $de \rightarrow en$ denotes that the model was trained using German documents and tested on English texts).

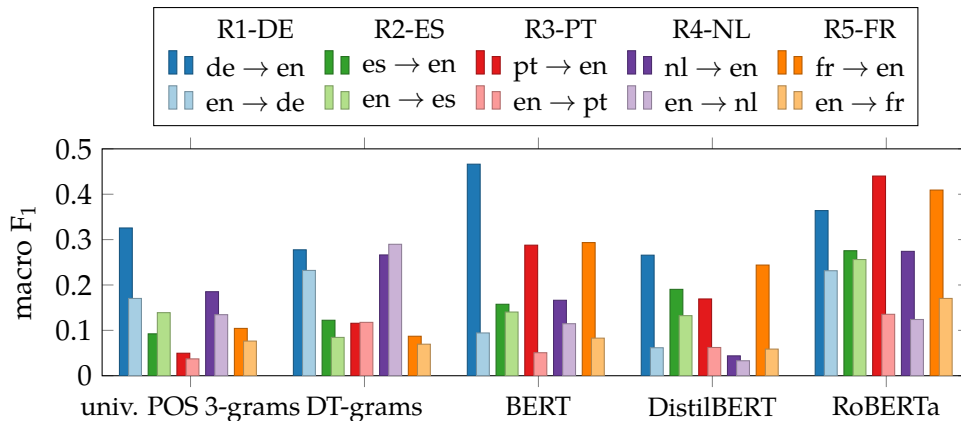


Figure 6.3.: Performance of the models on different language pairs and train/test directions. $de \rightarrow en$ denotes the evaluation scenario where the model is trained with the German documents and is tested with the English documents of the dataset.

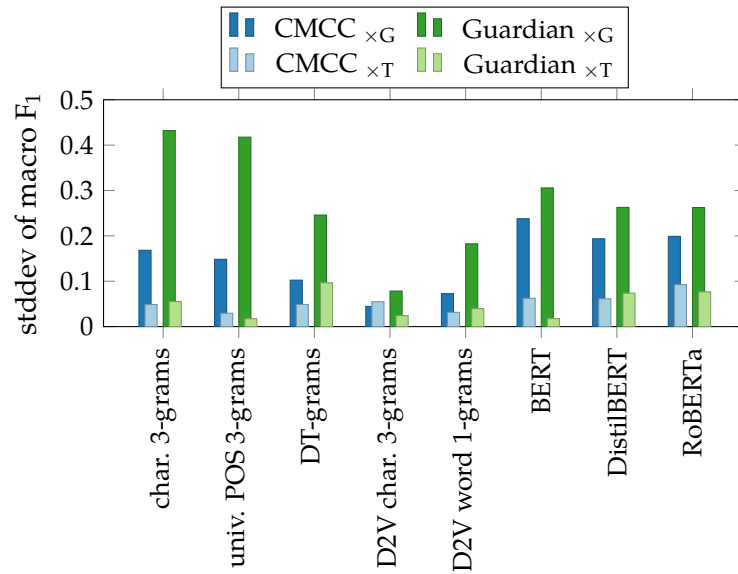


Figure 6.4.: Sensitivity of tested models to cross-genre (\times_G) and cross-topic (\times_T) splits. The y-axis shows the standard deviation of the F₁ score for all splits, high values indicate that the model performed well on some topics/genres and bad on others.

The performance of the models generally differs across different pairs, which suggests that any cross-language classification approach should use as many language pairs as possible to generalize well. However, cross-language datasets are difficult to compile, as authors writing in more than one language are sparse (cf. Chapter 3).

In general, but especially for the pre-trained language models, the figure also displays that the models perform better when they are fine-tuned using the non-English documents. This is especially interesting as for each dataset, the set of English documents is larger (cf. column \overline{dpa} of Table 3.8), suggesting the opposite of the observed results. This suggests that the choice of which language is used for training is an important choice that must be considered and reported by cross-language attribution studies, and can be more important than obtaining more training data for a specific language.

Sensitivity to Genre and Topic

From Table 6.8, several conclusions can be drawn from the results of the cross-topic/genre datasets. Firstly, we can confirm that cross-genre classification is in general harder than cross-topic classification. Where explicit previous

assumptions in this regard use single models and datasets [86, 4], we affirm this finding with multiple models and datasets. As a single exception, the character 3-grams show a higher performance on the cross-genre version of the CMCC dataset compared to the cross-topic variant.

The table also clearly reflects the difficulty that cross-genre situations impose on the pre-trained language models, which otherwise excel in the cross-topic splits.

Figure 6.4 shows the standard deviation of the F_1 score across the different topics and genres in the CMCC and Guardian datasets. Hence, high values mean that the models perform differently for the topics or genres in the dataset. The figure displays that most models are more sensitive to the genre of the text than they are to the topic, consistently over both CMCC and Guardian datasets. The Doc2Vec model with character 3-grams has a low overall prediction score (cf. Table 6.8), and shows this effect to a smaller degree.

Note that this does not hold for the average performance over all splits (cf. Table 6.8): in general, the tested models are performing better on the cross-topic datasets, and do so more consistently for all topics compared to the cross-genre datasets. This result can't be seen from Table 6.8, and it means that for some cross-genre splits, some models may perform better than the average winner.

Aggregated Score Results

Table 6.9 shows the results of the aggregated scores discussed in Section 6.4 for all models tested in our experiments, while Table 6.10 shows the standard deviation of each model across the different splits of the respective score. The aim of this separation is to quickly provide an overview of the strengths and weaknesses that a model shows for specific aspects of the datasets. For example, for the models presented in this chapter, it is now more clearly detectable that the character 3-gram features are a very strong baseline. Measured by the overall average F_1 score, they outperform all other approaches by at least 6%. However, when looking at the separate aggregated scores, they are unsuited for the cross-language tasks (which is expected of character-based features).

Another overall interesting result is the relatively good performance of pre-trained language models, especially on the "mono" score. When comparing the scores of "mono" to "small", the language models' performance drops

from 0.82 macro F_1 to 0.45, while the character-based 3-grams only drop from 0.83 to 0.62. Overall, this coincides with the results shown in Table 6.2, indicating that these language models have more difficulties with small numbers of training documents compared to other models.

Compared to other approaches, DT-grams are outperformed in each aggregated score. Only when looking at the `Redditsm` dataset separately, the features perform better than any other approach with 0.18 macro F_1 (cf. Table 6.8). As soon as the larger, non-balanced Reddit datasets are used, DT-grams are outperformed by RoBERTa (0.17 macro F_1 vs. 0.27).

DT-grams are also not effective in scenarios other than cross-language setups. For example, they are outperformed in every other category by the simpler universal POS tag 3-grams, also in a mixed-language scenario.

This indicates that the DT-gram features are highly specialized for the scenario of cross-language authorship attribution, and can't compete with other features for different scenarios of authorship attribution. It remains an open question whether the DT-grams may be useful as a supplementary feature in combination with other approaches. Since DT-grams inherently capture syntactic features, combinations using content-based features such as language models are an interesting option left for future research.

The standard deviations of the respective models' performances across the different scores, listed in Table 6.10, further show that the character n -grams not only perform well in many dataset combinations but do so consistently. Only the Doc2Vec models show lower deviations, but we interpret this as more of an artifact of their generally low performance, which is visible in Table 6.9.

The pre-trained language models show promising results for authorship attribution in summary, especially in the unexpected case of cross-language classification. Higher standard deviations indicate that these models are more prone to overfitting.

6.6. Conclusion and Discussion

In this chapter, a benchmark of various authorship attribution datasets is presented. It focuses on including as many different aspects as possible so that the strengths and weaknesses of text classification features and models can be interpreted intuitively. Thereby, a set of aggregated scores combines the re-

model	mono	small	mixed-language	cross-topic	cross-genre	cross-language	average
char. 3-grams	0.84	0.62	0.49	0.70	0.65	0.07	0.56
u.POS 3-grams	0.77	0.51	0.38	0.65	0.48	0.14	0.49
DT-grams	0.70	0.38	0.28	0.50	0.30	0.17	0.39
Doc2Vec char	0.25	0.19	0.38	0.26	0.23	0.04	0.22
Doc2Vec word	0.48	0.25	0.22	0.36	0.31	0.06	0.28
BERT	0.82	0.45	0.33	0.66	0.35	0.17	0.46
DistilBERT	0.82	0.43	0.39	0.61	0.33	0.12	0.45
RoBERTa	0.82	0.42	0.42	0.70	0.33	0.25	0.49

Table 6.9.: Aggregated macro F_1 scores reached by the models tested in our experiments.

model	mono	small	mixed-language	cross-topic	cross-genre	cross-language	average
char. 3-grams	0.19	0.05	0.11	0.12	0.23	0.06	0.13
u.POS 3-grams	0.22	0.09	0.29	0.14	0.21	0.09	0.17
DT-grams	0.24	0.01	0.16	0.15	0.13	0.08	0.13
Doc2Vec char	0.08	0.05	0.23	0.14	0.06	0.04	0.10
Doc2Vec word	0.11	0.04	0.10	0.14	0.11	0.05	0.09
BERT	0.22	0.14	0.19	0.17	0.24	0.13	0.18
DistilBERT	0.22	0.13	0.19	0.18	0.20	0.09	0.17
RoBERTa	0.23	0.21	0.13	0.14	0.21	0.11	0.17

Table 6.10.: Aggregated standard deviations of macro F_1 scores reached by the models tested in our experiments.

sult of different datasets to easily show the performance on different aspects of the texts.

Results from various different classification models run against the benchmark show that character n -grams are a strong baseline for a multitude of different datasets and scenarios, including cross-topic or genre classification tasks. In the context of this thesis, a further interesting result is the good performance of pre-trained language models on cross-language classification tasks exceeding a specific amount of training data, while the DT-grams feature performs better in experiments where the number of training documents is limited. In general, DT-grams are an efficient feature for a highly specialized use case of small-scale CLAA and show no promising results for deviating scenarios.

Fundamentally, the benchmark is intended to be used outside the limited scope of cross-language classification and is designed to be easily extensible, both in terms of adding further datasets and also adding further aggregated scores. For example, scores for specific genres of text (i.e., how well does a model predict *social media* texts compared to *emails*?) or grouping by temporal aspects (i.e., news articles from the 1970s vs. 2020s) could lead to new insights in this field. Therefore, the source code is made public as part of the underlying publication²⁵. Unfortunately, several of the datasets used in this version of the benchmark are not freely available, including the CMCC or Guardian datasets. In these cases, the authorization of the authors of the original publications must be obtained in order to use the dataset for any experiments. They are included in this work nevertheless, as they have unique characteristics which make them highly suitable for the purpose of this benchmark.

Summarized, the set of evaluation methods proposed in this chapter is suitable to analyze the performance of text classification setups for different aspects of datasets and are therefore able to answer the last research question of this thesis: *How can approaches in authorship attribution be evaluated in a way that shows their strengths and weaknesses of dataset aspects?*. As for the second part of the research question, *How do the features of RQ2 [i.e., DT-grams] compare to existing solutions?*, the results of this chapter show that DT-grams are efficient in a specialized scenario of small-scale cross-language authorship attribution, but don't show the capability of generalization to other scenarios.

²⁵<https://github.com/bmuraue/autbench>

Evaluating DT-grams in other Text Classification Fields

In this chapter, the performance of DT-grams on text classification tasks other than authorship attribution is analyzed, including experiments on authorship profiling and conspiracy detection, both of which are fields for which the feature was not intentionally developed. The purpose of these experiments is to measure the generalizability of DT-grams outside the context of authorship attribution.

7.1. Introduction

Knowing the capability of a text classification feature to perform well in different tasks is useful information when designing experiments from scratch, and helps to select useful baselines to compare against. For example, character n -grams are a versatile feature and provide solid baselines for many different purposes ranging from spam detection [39, 24] to aiding optical character recognition development [18]. Likewise, pre-trained language models such as BERT [17] and variations thereof have been used in a similarly heterogeneous set of NLP tasks. In this chapter, an experimental approach is used to obtain results for DT-gram performances in different text classification fields than authorship attribution.

Compared to Chapter 6, this means that this chapter goes one step further and explores two text classification fields only loosely related to attribution. Thereby, the first experiment analyzes *authorship profiling*, a task similar to attribution where stylistic characteristics are detected not for single persons (authors), but for groups of writers that share a common property like gender or age.

Thereafter, the results of using DT-grams in conspiracy detection are presented, where text messages are classified depending on whether they contain indications of including a set of conspiracies. Semantically, this task can be described as even farther away from authorship attribution, as it no longer intuitively can be explained using a writer's *style*.

7.2. Authorship Profiling

Authorship profiling is the task of predicting a property or aspect of a document's author. The applications for this task are similar to those of authorship attribution, for example, to reduce a pool of suspected authors if some properties of an author are known beforehand. Profiling and attribution differ in the important point that for profiling, the outcome variable is not necessarily tied to single persons, but rather collections of persons that share the same property or aspect. The influence of this difference will become important later in this section. Like attribution, the practical fields of use for automatic authorship profiling are limited due to the low overall accuracy of the procedures [84]. Even when the set of candidates is relatively small, manual analysis of the evidence is still an open and challenging issue [1, 23].

In this section, the dataset of the 2015 PAN shared task for authorship profiling is used to evaluate the suitability of DT-grams for this task.

Dataset and Experiment Setup

The PAN 2015 shared task for authorship profiling [68] consists of 4 sub-problems in English, Italian, Spanish and Dutch, respectively. For each task, a pre-defined set of training and testing documents is available. In this section, the evaluations are performed using the *gender* and *age group* target field, making it a profiling task. However, only the *gender* field is available for all languages, and the *age group* variable is only available for the English and Spanish sub-problems. The dataset provides several psychological traits as further possible target fields, which are disregarded in this experiment.

Table 7.1 shows the most important characteristics of the datasets.

The nature of the dataset allows for different evaluation strategies in terms of the cross-language nature of the DT-grams feature. In total, we conduct three different sets of experiments:

	documents	authors	avg. docs/author
total			
	59,366	622	95
age groups			
18-24	14,822	154	96
25-34	19,813	208	95
35-49	7,627	82	93
50-XX	3,570	38	93
XX-XX	13,534	140	96
gender			
F	30,106	311	96
M	29,260	311	94

Table 7.1.: The PAN 2015 Profiling Dataset. The ‘XX-XX’ age group denotes that this information is not available for the respective document.

1. We perform *single-language profiling* by analyzing each sub-problem independently and use language-specific POS tags as node representation for the DT-grams.
2. By using all documents from all subproblems’ training parts, we conduct *mixed-language* profiling. In order for this strategy to work, we must use *universal* POS tags.
3. Lastly, a *cross-language* setup is evaluated by using the *leave-one-language-out* evaluation strategy: For each language in the dataset, the three respective other languages are used for training, while the left-out language is used for testing. This way, the languages of the training and testing documents don’t overlap, resulting in a classification setup comparable to the experiments of Section 5.

For all evaluations, the prediction performance of the DT-grams (with $\sigma=s10$, $\delta_{anc}=2$, and $\delta_{sib}=3$) is compared to two standard approaches in this field: character n -grams and the multilingual version of the BERT classifier.

Results

Figure 7.1 shows the results of the three profiling experiments predicting the *gender* of the authors. It is clear that the DT-grams feature does not compete

with the two baseline approaches tested in this section, and provides inferior results in all three cases. In the single- and mixed language setups, the DT-grams are outperformed by simple character n -grams, and both are exceeded by mBERT in the cross-language setting.

When reducing the number of training documents to 10, a picture similar to the experiments in Section 6.5 converges, as the language model's and the character n -gram performance drops, whereas the DT-grams are not affected as much. However, in authorship profiling, this scenario is less relevant than is the case with authorship attribution, as it is more feasible to obtain more training documents from authors of a specific *gender* than it is to obtain more documents from a specific *author*. This means that scenarios with such a reduced training set are less likely unless a very specific target property is predicted.

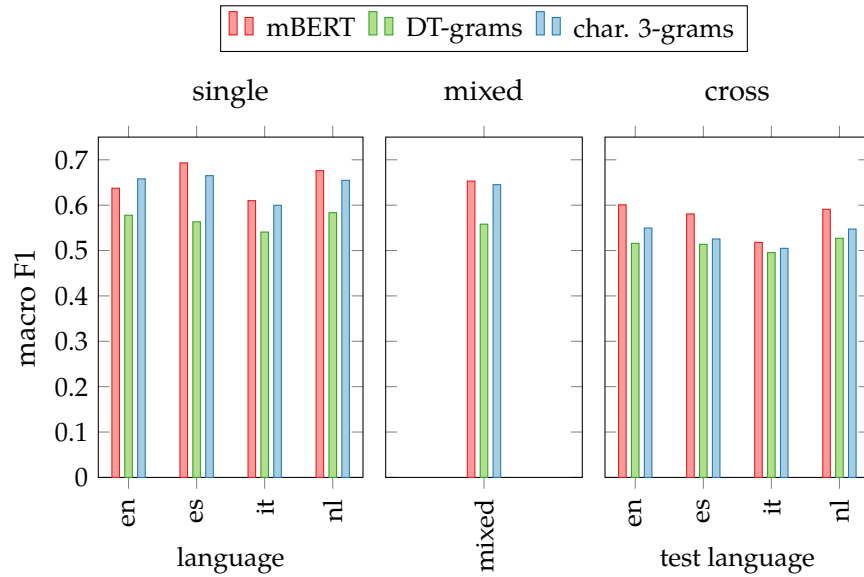
The results of the experiments using the *age group* as the prediction target (which is only available for the English and Spanish sub-problems) paint a similar picture, although the cross-language scenario is not dominated by mBERT as much as was the case with the gender prediction.

Summarized, the evaluation experiments suggest that the DT-grams feature is not a suitable candidate for profiling. Data-intensive models like mBERT show a superior performance out-of-the-box, and unlike authorship attribution, it is easier to increase the amount of training data for specific traits. However, for special cases where the number of training documents is small and difficult to increase, a niche use case might exist.

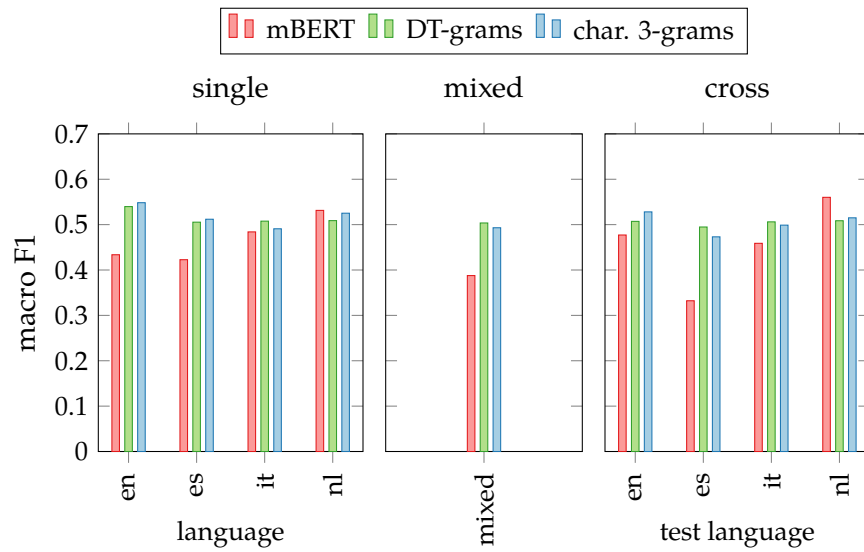
7.3. Conspiracy Detection²⁶

The MediaEval 2021 workshop on the shared task "FakeNews: Corona Virus and Conspiracies Multimedia Analysis Task" includes multiple text classification problems that aim to increase the understanding of detecting fake news messages on Twitter. It proposes three different classification tasks while using the same textual data for each of them:

²⁶Results and contents of this section are based on and partially reused from the paper: Manfred Moosleitner and Benjamin Murauer: *On the Performance of Different Text Classification Strategies on Conspiracy Classification in Social Media*. In CEURS Working Notes Proceedings of the MediaEval 2021 Workshop. CEUR-WS.org, 2022 preliminary proceedings.



(a) Using all training documents



(b) Using 10 training documents

Figure 7.1.: Results of profiling Experiments using *gender* as prediction target, using all (a) and a limited number (b) of training documents.

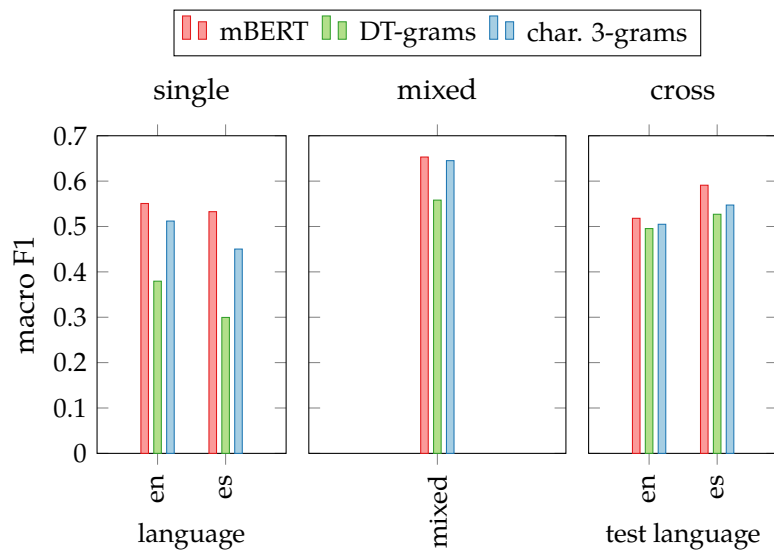


Figure 7.2.: Results of profiling experiments using *age_group* as prediction target.

1. Misinformation detection: each document belongs to one of three outcome classes ("promotes/supports conspiracy", "discusses conspiracy" or "non-conspiracy"), and the model must predict the correct class.
2. Conspiracy theory recognition: each document contains one or more conspiracies from a provided list (Suppressed cures, Behaviour and Mind Control, Antivax, Fake virus, Intentional Pandemic, Harmful Radiation or Influence, Population reduction, New World Order, and Satanism), and the model must predict which of the theories are contained in the document.
3. Combined detection: a combination of the previous two tasks - the model must detect which conspiracies are mentioned in the document, and in which way.

The main goal of participating in this task was to provide an evaluation of DT-grams on an obviously content-reliant classification task, where the writing style of the author is not obviously helpful in the decision process.

7.3.1. Dataset and Methodology

The dataset provided by the task organizers consists of ~2k tweets [72], whereas 1,554 were provided to the task participants as the development dataset, and 266 were used to evaluate the final solutions. The Twitter messages are limited to a length of 240 characters, and no meta-information is included in the data to ensure that any classification effort is performed on the textual data alone.

We performed classification experiments with different models that have been presented in this thesis:

- tf/idf normalized frequencies of character and word n -grams in combination with different classification models (linear SVM, extra randomized trees, and multinomial naive bayes)
- tf/idf normalized frequencies of DT-grams with a linear SVM
- pre-trained language models (BERT, RoBERTa, DistilBERT)

Thereby, various hyperparameters were tested in a grid-search approach, which are listed in Table 7.2.

As classification metric, the task organizers selected *Matthew’s correlation coefficient* (MCC), which is defined as (cf. Section 2.5 for definitions):

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}$$

Compared to the previously used F_1 score, one difference between the two metrics is that the MCC score also incorporates the number of *true negatives*,

parameter	tested Values
n -gram size	1, 2, ..., 10
n -grams max. features	unlimited, 1000
lowercase text	true, false
DT-gram word repr.	universal POS tag, English POS tag
BERT model	RoBERTa, DistilBERT, BERT base
trees	100, 250, 500, 750, 1000, 2000, ..., 5000

Table 7.2.: Hyperparameters tested in grid-search.

features	char 6-grams tf-idf	plain text sequence	word 1-grams tf-idf	dt-gram tf-idf
model	ET	BERT	SVM	MNNB
task 1	0.2852	0.3184	0.2228	0.1201
task 2	0.2086	0.3624	0.2879	0.0000
task 3	0.1993	0.3347	0.2316	-0.0028

Table 7.3.: Evaluation results measured with Matthew’s correlation coefficient.

while the F_1 score does not. This difference becomes more important on imbalanced datasets, where varying importance of true negative samples may lead to a different interpretation of results. However, no scientific consensus is reached on providing definitive answers on whether the F_1 or MCC score should be preferred when dealing with imbalanced datasets [13, 104].

7.3.2. Results

Table 7.3 shows the results of the tested models on the specific tasks. The pre-trained language model BERT achieves the highest scores in all three tasks, whereas it is clearly visible that the grammar-based DT-grams are not able to capture any relevant information, and are not suitable for this heavily content-based classification task.

This intuition is undermined when looking at the weights of the SVM model that it applies to the word 1-grams, where words that are clearly related to the specific conspiracy topics are given high weights (cf. Figure 7.3). The figure also demonstrates that further optimizations are possible. For example, the two top positive terms for "Suppressed Cures" are "microchip" and "microchips", which could be combined using stemming or lemmatization.

7.4. Conclusion

This chapter demonstrated that DT-grams are generally ill-suited for text classification tasks other than authorship attribution. Two different text classification scenarios are used to compare DT-grams to other approaches used in the respective areas: Authorship profiling and conspiracy detection. While the DT-grams’ performance was comparable to (but never exceeding) the other

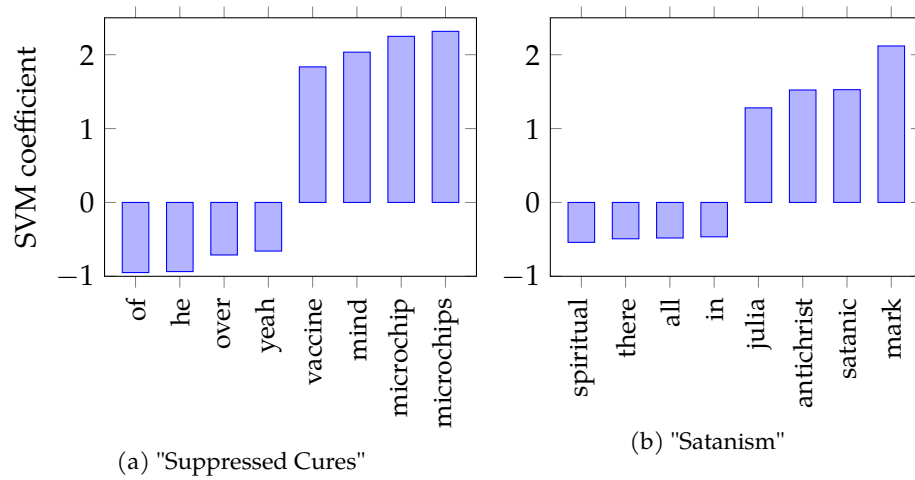


Figure 7.3.: Top 4 positive and negative coefficients of the classes "Suppressed Cures" and "Satanism".

approaches in the profiling task, the results of the conspiracy detection task show that syntactic features are barely exceeding the random baseline.

The latter result is somewhat expectable since the word's content is intuitively an important feature for the task of detecting conspiracies. Further research using ensemble methods with additional content-based features may provide further context in this regard.

In the profiling experiment, on the other hand, DT-grams fail to capture expressive common stylistic features of different genders and age groups. This indicates that the profiling task is more centered around the stylistic choices of words and content rather than grammatical choices.

Conclusion

Stylistic analysis of documents and determining which features are expressive for specific authors is a difficult task that has been the focus of many studies in natural language processing recently. Thereby, the scenario of cross-language attribution problems involving authors writing in multiple languages was addressed only marginally. In this thesis, several contributions to this field are presented.

Cross-Language Authorship Datasets

A fundamental resource for any type of research are datasets that are used for developing and evaluating models. In the field of cross-language authorship research, a large gap was filled by the work presented in this thesis: Previously, only human-translated datasets existed that included authors that wrote documents in one language, and translated works were used as a source for text in different languages. In this thesis, an approach is presented that uses a vast source of social media comments to compose datasets with different properties. Concretely, this thesis presents five datasets consisting of cross-language authors writing in English as well as German, Dutch, Spanish, Portuguese and French. This lays the foundation for the second contribution: the development of a novel family of features for authorship classification.

DT-Grams

Natural language is more than a mere collection of tokens with their individual semantic meaning, and using syntactic information to describe the writing styles of authors has been used in previous research in different tasks. In this thesis, the DT-grams feature family is presented, which represents a language-independent version of syntactic features. Thereby, the sentences of texts are represented as dependency graph structures, and each word in the sentence is

represented by a language-independent part-of-speech tag. By sampling subsections of the graph and counting their frequencies, profiles for authors can be constructed. DT-grams have many hyperparameters that can be tweaked in different ways, and recommended values for social media authorship attributions are presented in this thesis.

Experiments have demonstrated that this type of feature is especially powerful and useful in small datasets, where other state-of-the-art models have trouble with learning enough information from the limited amount of training data. Thereby, combining DT-grams with current machine translation approaches outperforms other widely used methods for this type of data.

Authorship Attribution Benchmark

Comparing the performance of the DT-grams to other authorship attribution methods in the field is difficult, as many approaches limit evaluation experiments to a selected number of datasets. Therefore, this thesis presents a comprehensive collection of datasets, representing a systematic benchmark for authorship attribution. It includes many established datasets widely used in the field, as well as the social media cross-language datasets presented in this thesis. Moreover, it includes strategies to evaluate a model's performance based on different textual aspects such as document length or the number of documents per author. The aim of the benchmark is to provide a more complete image of how well a model performs, and how sensitive that performance is to a dataset's specific properties.

The results of the benchmark reveal several insights, including the strength of pre-trained language models for authorship attribution tasks on datasets with sufficient training data, or how much data is required for them to work at all. Further, the benchmark demonstrates that the performance of DT-grams is outmatched by many approaches in traditional single-language scenarios. This is also true for tasks other than authorship attribution, where experiments in the related tasks of authorship profiling and fake news detection show poor results for the approach. However, several promising outlooks are yet to be researched in the context of DT-grams, like exploring the possibilities of combining DT-grams with other features.

In summary, the research questions stated in Chapter 1 can be answered as follows:

RQ1: *How can datasets be obtained that are suitable for CLAA?*

The use of social media comments allows composing datasets from multilingual authors without the need for human translation.

RQ2: *Which language-independent syntax-based features are a viable choice for a classification feature for CLAA?*

By representing sentences with a combination of universal dependencies and universal part-of-speech tags and creating author profiles based on the frequency of the substructures of the dependency graph yields a machine learning feature that is powerful in small-scale social media datasets, where the amount of training data is limited.

RQ3: *How can approaches in authorship attribution be evaluated in a way that shows their strengths and weaknesses of dataset aspects, and how do the features of RQ2 compare to existing solutions?*

The intuitive solution of combining many different datasets in combination with aggregating evaluation results based on properties of the respective datasets leads to a broadly applicable benchmark that better shows the strengths and weaknesses of a model with respect to those properties.

Bibliography

- [1] Shlomo Argamon. Computational Forensic Authorship Analysis: Promises and Pitfalls. *Language and Law / Linguagem e Direito*, 5(2):7–37, 2018.
- [2] Shlomo Argamon and Shlomo Levitan. Measuring the usefulness of function words for authorship attribution. In *Proceedings of the Joint Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing*, pages 1–3, 2005.
- [3] Nikolaus Augsten, Michael Böhlen, and Johann Gamper. PQ-Gram Distance Between Ordered Labeled Trees. *ACM Transactions on Database Systems*, pages 1–35, 2010.
- [4] Georgios Barlas and Efstathios Stamatatos. Cross-Domain Authorship Attribution Using Pre-trained Language Models. In *IFIP Advances in Information and Communication Technology*, pages 255–266, 2020. doi: [10.1007/978-3-030-49161-1_22](https://doi.org/10.1007/978-3-030-49161-1_22).
- [5] Dasha Bogdanova and Angeliki Lazaridou. Cross-language authorship attribution. In *Ninth International Conference on Language Resources and Evaluation (LREC'2014)*, pages 2015–2020, 2014.
- [6] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information, July 2016. arXiv: [1607.04606](https://arxiv.org/abs/1607.04606).
- [7] Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. TIGER: Linguistic Interpretation of a German Corpus. *Journal of Language and Computation*, 2:597–620, 2004.
- [8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam,

- Girish Sastry, Amanda Aspell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners, May 2020. [arXiv:2005.14165](https://arxiv.org/abs/2005.14165).
- [9] Paweł Budzianowski and Ivan Vulić. Hello, It's GPT-2 – How Can I Help You? Towards the Use of Pretrained Language Models for Task-Oriented Dialogue Systems, July 2019. [arXiv:1907.05774](https://arxiv.org/abs/1907.05774).
- [10] Serhiy Bykh and Detmar Meurers. Exploring syntactic features for native language identification: A variationist perspective on feature encoding and ensemble optimization. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING'2014)*, pages 1962–1973, August 2014.
- [11] Nicola Cancedda, Eric Gaussier, Cyril Goutte, and Jean Michel Renders. Word sequence kernels. *The Journal of Machine Learning Research*, 3:1059–1082, 2003.
- [12] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, aug 2016. [doi:10.1145/2939672.2939785](https://doi.org/10.1145/2939672.2939785).
- [13] Davide Chicco and Giuseppe Jurman. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1), jan 2020. [doi:10.1186/s12864-019-6413-7](https://doi.org/10.1186/s12864-019-6413-7).
- [14] Noam Chomsky. *Syntactic Structures*. Mouton, 1957.
- [15] Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4585–4592, 2014.
- [16] Pierre A. Devijver. *Pattern recognition*. Prentice/Hall International, 1982.
- [17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language un-

derstanding, 2018. URL: <http://arxiv.org/abs/1810.04805>.

- [18] Shrey Dutta, Naveen Sankaran, K. Pramod Sankar, and C.V. Jawahar. Robust Recognition of Degraded Documents Using Character N-Grams. In *2012 10th IAPR International Workshop on Document Analysis Systems*. IEEE, mar 2012. doi:10.1109/das.2012.76.
- [19] Maciej Eder. Style-markers in authorship attribution : a cross-language study of the authorial fingerprint. *Studies in Polish Linguistics*, 6(1):99–114, 2011.
- [20] Maciej Eder. Does size matter? Authorship attribution, small samples, big problem. *Digital Scholarship in the Humanities*, 30(2):167–182, 2013.
- [21] Maciej Eder. Mind your corpus: systematic errors in authorship attribution. *Literary and Linguistic Computing*, 28(4):603–614, jul 2013. doi:10.1093/llc/fqt039.
- [22] Maciej Eder and Jan Rybicki. Do birds of a feather really flock together, or how to choose training samples for authorship attribution. *Literary and Linguistic Computing*, 28(2):229–236, 2012.
- [23] Eilika Fobbe. Text-Linguistic Analysis in Forensic Authorship Attribution. *International Journal of Language and Law*, 9:93–114, 2020. doi:10.14762/jll.2020.093.
- [24] Donato Hernández Fusilier, Manuel Montes-y Gómez, Paolo Rosso, and Rafael Guzmán Cabrera. Detection of Opinion Spam with Character n-grams. In *Computational Linguistics and Intelligent Text Processing*, pages 285–294. Springer International Publishing, 2015. doi:10.1007/978-3-319-18117-2_21.
- [25] A. M. Garcia and J. C. Martin. Function Words in Authorship Attribution Studies. *Literary and Linguistic Computing*, 22(1):49–66, nov 2006. doi:10.1093/llc/fql048.
- [26] Jade Goldstein-Stewart, Kerri Goodwin, Roberta Sabin, and Ransom Winder. Creating and Using a Correlated Corpus to Glean Communicative Commonalities. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. European Language Resources Association (ELRA), May 2008.
- [27] Jade Goldstein-Stewart, Ransom Winder, and Roberta Sabin. Person Identification from Text and Speech Genre Samples. In *Proceedings of*

- the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 336–344, Athens, Greece, March 2009. Association for Computational Linguistics.
- [28] Jack Grieve. Quantitative authorship attribution: An evaluation of techniques. *Literary and Linguistic Computing*, 22:251–270, 2007.
- [29] David G. Hays. Grouping and dependency theories. In *Proceedings of the National Symposium on Machine Translation*, pages 258–266, 1961.
- [30] David I. Holmes. Authorship Attribution. *Computers and the Humanities*, 28(2):87–106, 1994.
- [31] Warren Hope. *The Shakespeare Controversy*. McFarland, second edition edition, 2009.
- [32] Eva Lorenzo Iglesias, A Seara Vieira, and Lourdes Borrajo. An HMM-based over-sampling technique to improve text classification. *Expert Systems with Applications*, 40(18):7184–7192, 2013.
- [33] Radu Tudor Ionescu and Marius Popescu. Can string kernels pass the test of time in Native Language Identification? In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 224–234. Association for Computational Linguistics, September 2017.
- [34] Radu Tudor Ionescu, Marius Popescu, and Aoife Cahill. Can characters reveal your native language? A language-independent approach to native language identification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP’2014)*, pages 1363–1373. Association for Computational Linguistics, October 2014.
- [35] Radu Tudor Ionescu, Marius Popescu, and Aoife Cahill. String kernels for native language identification: Insights from behind the curtains. *Computational Linguistics*, 42(3):491–525, 2016.
- [36] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972. [doi:10.1108/eb026526](https://doi.org/10.1108/eb026526).
- [37] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. Marian: Fast Neural Machine Translation in

C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121. Association for Computational Linguistics, July 2018.

- [38] Neli Kalcheva, Milena Karova, and Ivaylo Penev. Comparison of the accuracy of SVM kernel functions in text classification. In *2020 International Conference on Biomedical Innovations and Applications (BIA)*, pages 141–145, 2020.
- [39] Ioannis Kanaris, Konstantinos Kanaris, and Efstathios Stamatatos. Spam Detection Using Character N-Grams. In *Advances in Artificial Intelligence*, pages 95–104. Springer Berlin Heidelberg, 2006. doi:[10.1007/11752912_12](https://doi.org/10.1007/11752912_12).
- [40] Mike Kestemont, Michael Tschuggnall, Efstathios Stamatatos, Walter Daelemans, Günther Specht, Benno Stein, and Martin Potthast. Overview of the Author Identification Task at PAN 2018: Cross-domain Authorship Attribution and Style Change Detection. In *Working Notes Papers of the CLEF 2018 Evaluation Labs*, September 2018. doi:[10.5281/zenodo.3737849](https://doi.org/10.5281/zenodo.3737849).
- [41] Phillip Keung, Yichao Lu, and Vikas Bhardwaj. Adversarial Learning with Contextual Embeddings for Zero-resource Cross-lingual Classification and NER, 2019. arXiv:[1909.00153](https://arxiv.org/abs/1909.00153).
- [42] Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Computational Methods in Authorship Attribution. *Journal of the American Society for Information Science and Technology*, 60(1):9–26, jan 2009. doi:[10.1002/asi.20961](https://doi.org/10.1002/asi.20961).
- [43] Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Authorship attribution in the wild. *Language Resources and Evaluation*, 45(1):83–94, jan 2011. doi:[10.1007/s10579-009-9111-2](https://doi.org/10.1007/s10579-009-9111-2).
- [44] Moshe Koppel, Jonathan Schler, Shlomo Argamon, and Eran Messeri. Authorship attribution with thousands of candidate authors. In *Proceedings of the 29th Conference on Research and Development in Information Retrieval (SIGIR'2006)*. ACM Press, 2006. doi:[10.1145/1148170.1148304](https://doi.org/10.1145/1148170.1148304).
- [45] Henry Kucera and Francis W. Nelson. *Computational Analysis of Present-Day American English*. Brown University Press, 1967.
- [46] Quoc Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on*

- Machine Learning*, pages 1188–1196. PMLR, 2014.
- [47] David D. Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(Apr):361–397, 2004.
- [48] Gang Liu and Jiabao Guo. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing*, 337:325–338, 2019.
- [49] Rui Liu, Junjie Hu, Wei Wei, Zi Yang, and Eric Nyberg. Structural Embedding of Syntactic Trees for Machine Comprehension. *CoRR*, abs/1703.00572, 2017. [arXiv:1703.00572](https://arxiv.org/abs/1703.00572).
- [50] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. July 2019. [arXiv:1907.11692](https://arxiv.org/abs/1907.11692).
- [51] Zhi Liu. Reuter 50-50 Dataset. National Engineering Research Center for E-Learning Technology China, 2011. URL: https://archive.ics.uci.edu/ml/datasets/Reuter_50_50.
- [52] Marisa Llorens and Sarah Jane Delany. Deep level lexical features for cross-lingual authorship attribution. In *Proceedings of the first Workshop on Modeling, Learning and Mining for Cross/Multilinguality*, pages 16–25. Dublin Institute of Technology, 2016.
- [53] Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2(Feb):419–444, 2002.
- [54] Kim Luyckx and Walter Daelemans. The effect of author set size and data size in authorship attribution. *Literary and Linguistic Computing*, 26(1):35–55, 2011. [doi:10.1093/llic/fqq013](https://doi.org/10.1093/llic/fqq013).
- [55] Christopher Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, May 1999.
- [56] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, June 1993.

-
- [57] Thomas Corwin Mendenhall. The Characteristic Curve of Composition. *Science*, 9(214):237–249, 1887.
- [58] Rohith Menon and Yejin Choi. Domain independent authorship attribution without domain adaptation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 309–315, 2011.
- [59] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space, January 2013. [arXiv:1301.3781](#).
- [60] Frederick Mosteller and David L. Wallace. Inference in an Authorship Problem. *Journal of the American Statistical Association*, 58(302):275, jun 1963. [doi:10.2307/2283270](#).
- [61] Benjamin Murauer and Günther Specht. Generating Cross-Domain Text Classification Corpora from Social Media Comments. In *Proceedings of the 20th International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF'2019)*, pages 114–125, 2019. [doi:10.1007/978-3-030-28577-7_7](#).
- [62] Benjamin Murauer and Günther Specht. DT-grams: Structured Dependency Grammar Stylometry for Cross-Language Authorship Attribution. In *Proceedings of the 32nd GI-Workshop Grundlagen von Datenbanksysteme (GoDB'21)*, 2021.
- [63] Benjamin Murauer, Michael Tschuggnall, and Günther Specht. On the Influence of Machine Translation on Language Origin Obfuscation. June 2018. [arXiv:2106.12830](#).
- [64] Arvind Narayanan, Hristo Paskov, Neil Zhenqiang Gong, John Bethencourt, Emil Stefanov, Eui Chul Richard Shin, and Dawn Song. On the feasibility of internet-scale author identification. In *2012 IEEE Symposium on Security and Privacy*. IEEE, may 2012. [doi:10.1109/sp.2012.46](#).
- [65] Joakim Nivre. Dependency Parsing. *Language and Linguistics Compass*, 4(3):138–152, mar 2010. [doi:10.1111/j.1749-818x.2010.00187.x](#).
- [66] Joakim Nivre, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Masayuki Asahara, Luma Ateyah, Mohammed Attia,

- Aitziber Atutxa, Liesbeth Augustinus, et al. Universal Dependencies 2.1, 2017. URL: <https://universaldependencies.org/>.
- [67] Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th Int. Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, 2016.
- [68] Francisco Manuel Rangel Pardo, Fabio Celli, Paolo Rosso, Martin Potthast, Benno Stein, and Walter Daelemans. Overview of the 3rd Author Profiling Task at PAN 2015. In L. Cappellato, N. Ferro, J. Gareth, and E. San Juan, editors, *CEUR Workshop Proceedings*, volume 1391, 2015.
- [69] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.
- [70] Shanta Phani, Shibamouli Lahiri, and Arindam Biswas. Authorship attribution in bengali language. In *Proceedings of the 12th International Conference on Natural Language Processing*, pages 100–105, 2015.
- [71] Philipp Pobitzer. *Design and Evaluation of Different pq-gram Variants for Grammar-Based Text Analysis*. Bachelor thesis, Universität Innsbruck, Department of Computer Science, 2017.
- [72] Konstantin Pogorelov, Daniel Thilo Schroeder, Stefan Brenner, and Johannes Langguth. FakeNews: Corona Virus and Conspiracies Multimedia AnalysisTask at MediaEval 2021. *CEURS Working Notes Proceedings of the MediaEval 2021 Workshop*, 2022.
- [73] Marius Popescu and Cristian Grozea. Kernel methods and string kernels for authorship analysis. In *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes*, volume 1178. CEUR-WS.org, 2012.
- [74] Martin Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 2006. doi:10.1108/00330330610681286.
- [75] Neha Raghuvanshi and Jaikumar M. Patil. A Brief Review on Sentiment Analysis. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*. IEEE, mar 2016. doi:10.1109/iceeot.2016.7755213.

-
- [76] Markus Sadeniemi, Kimmo Kettunen, Tiina Lindh-Knuutila, and Timo Honkela. Complexity of European Union Languages: A Comparative Approach. *Journal of Quantitative Linguistics*, 15(2):185–211, 2008. doi:[10.1080/09296170801961843](https://doi.org/10.1080/09296170801961843).
- [77] Manuel Sage, Pietro Cruciata, Raed Abdo, Jackie Chi Kit Cheung, and Yaoyao Fiona Zhao. Investigating the Influence of Selected Linguistic Features on Authorship Attribution using German News Articles. In *SwissText/KONVENS*, volume 2624, 2020.
- [78] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBert, a distilled version of bert: smaller, faster, cheaper and lighter, October 2019. [arXiv:1910.01108](https://arxiv.org/abs/1910.01108).
- [79] Upendra Sapkota, Steven Bethard, Manuel Montes, and Tamar Solorio. Not All Character N-grams Are Created Equal: A Study In Authorship Attribution. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human language Technologies*, pages 93–102, June 2015.
- [80] Upendra Sapkota, Tamar Solorio, Manuel Montes y Gómez, Steven Bethard, and Paolo Rosso. Cross-Topic Authorship Attribution: Will Out-Of-Topic Data Help? In *Proceedings of the 25th International Conference on Computational Linguistics (COLING'2014)*, pages 1228–1237, August 2014.
- [81] Yunita Sari, Mark Stevenson, and Andreas Vlachos. Topic or style? exploring the most useful features for authorship attribution. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 343–353. Association for Computational Linguistics, August 2018.
- [82] Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. Collaborative Inference of Sentiments from Texts. In *Proceedings of the 18th International Conference on User Modeling, Adaptation and Personalization (UMOD'2010)*, pages 195–206, 06 2010. doi:[10.1007/978-3-642-13470-8_19](https://doi.org/10.1007/978-3-642-13470-8_19).
- [83] Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernández. Syntactic Dependency-Based N-grams as Classification Features. 11:1–11, 2013. doi:[10.1007/978-3-642-37798-3_1](https://doi.org/10.1007/978-3-642-37798-3_1).

- [84] Rui Sousa-Silva. Computational Forensic Linguistics: An Overview of Computational Applications in Forensic Contexts. *Language and Law / Linguagem e Direito*, 5(2):118–143, 12 2018.
- [85] James Spencer and Gulden Uchyigit. Sentimentor: Sentiment Analysis of Twitter Data. In *Proceedings of the 1st International Workshop on Sentiment Discovery from Affective Data at European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 56–66, 2012.
- [86] Efstathios Stamatatos. On the Robustness of Authorship Attribution Based on Character N-Gram Features. *Journal of Law & Policy*, pages 421–439, 2013.
- [87] Efstathios Stamatatos. Authorship Attribution Using Text Distortion. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL'2017)*, pages 1138–1149. Association for Computational Linguistics, April 2017.
- [88] Lauren M. Stuart, Saltanat Tazhibayeva, Amy R. Wagoner, and Julia M. Taylor. Style features for authors in two languages. In *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*. IEEE, nov 2013. doi:[10.1109/wi-iat.2013.65](https://doi.org/10.1109/wi-iat.2013.65).
- [89] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to Fine-Tune BERT for Text Classification? In *Lecture Notes in Computer Science*, pages 194–206. Springer International Publishing, 2019. doi:[10.1007/978-3-030-32381-3_16](https://doi.org/10.1007/978-3-030-32381-3_16).
- [90] Michael Tschuggnall, Benjamin Murauer, and Günther Specht. Reduce & Attribute: Two-Step Authorship Attribution for Large-Scale Problems. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 951–960. Association for Computational Linguistics, November 2019. doi:[10.18653/v1/K19-1089](https://doi.org/10.18653/v1/K19-1089).
- [91] Michael Tschuggnall and Günther Specht. Detecting Plagiarism in Text Documents through Grammar-Analysis of Authors. In *15. GI-Fachtagung Datenbanksysteme für Business, Technologie und Web (BTW'2013)*, pages 241–259, 2013.
- [92] Michael Tschuggnall and Günther Specht. Countering Plagiarism by Exposing Irregularities in Authors' Grammar. In *Proceedings of the Euro-*

pean Intelligence and Security Informatics Conference, (EISIC'2013), pages 15–22. IEEE, 2013. doi:[10.1109/EISIC.2013.10](https://doi.org/10.1109/EISIC.2013.10).

- [93] Michael Tschuggnall and Günther Specht. Using Grammar-profiles to Intrinsically Expose Plagiarism in Text Documents. In Elisabeth Métais, Farid Meziane, Mohamad Saraee, Vijayan Sugumaran, and Sunil Vadera, editors, *Proceedings of the 18th International Conference on Applications of Natural Language to Information Systems (NLDB'2013)*, pages 297–302. Springer Berlin Heidelberg, 06 2013.
- [94] Michael Tschuggnall and Günther Specht. Enhancing Authorship Attribution By Utilizing Syntax Tree Profiles. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL'2014)*, volume 2, pages 195–199. Association for Computational Linguistics, 2014.
- [95] Michael Tschuggnall and Günther Specht. On the Potential of Grammar Features for Automated Author Profiling. *International Journal on Advances in Intelligent Systems*, 8(3):255–265, 2015.
- [96] C.J. van Rijsbergen. *Information Retrieval*. Butterworths, 2 edition, 1979.
- [97] Lawrence Venuti. *The translator's invisibility: A history of translation*. Routledge, 2008.
- [98] Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. TOD-BERT: Pre-trained Natural Language Understanding for Task-Oriented Dialogue, April 2020. [arXiv:2004.06871](https://arxiv.org/abs/2004.06871).
- [99] Shijie Wu and Mark Dredze. Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT, April 2019. [arXiv:1904.09077](https://arxiv.org/abs/1904.09077).
- [100] Kwan Yi and Jamshid Beheshti. A hidden Markov model-based text classification of medical documents. *Journal of Information Science*, 35(1):67–81, 2009.
- [101] Richong Zhang, Zhiyuan Hu, Hongyu Guo, and Yongyi Mao. Syntax Encoding with Application in Authorship Attribution. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'2018)*. Association for Computational Linguistics, 2018. doi:[10.18653/v1/d18-1294](https://doi.org/10.18653/v1/d18-1294).

- [102] Yunxiang Zhang and Zhuyi Rao. n-bilstm: Bilstm with n-gram features for text classification. In *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pages 1056–1059. IEEE, 2020.
- [103] Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling. November 2016. [arXiv:1611.06639](https://arxiv.org/abs/1611.06639).
- [104] Qiuming Zhu. On the performance of Matthews correlation coefficient (MCC) for imbalanced dataset. *Pattern Recognition Letters*, 136:71–80, 2020. [doi:10.1016/j.patrec.2020.03.030](https://doi.org/10.1016/j.patrec.2020.03.030).

Appendices

DT-grams

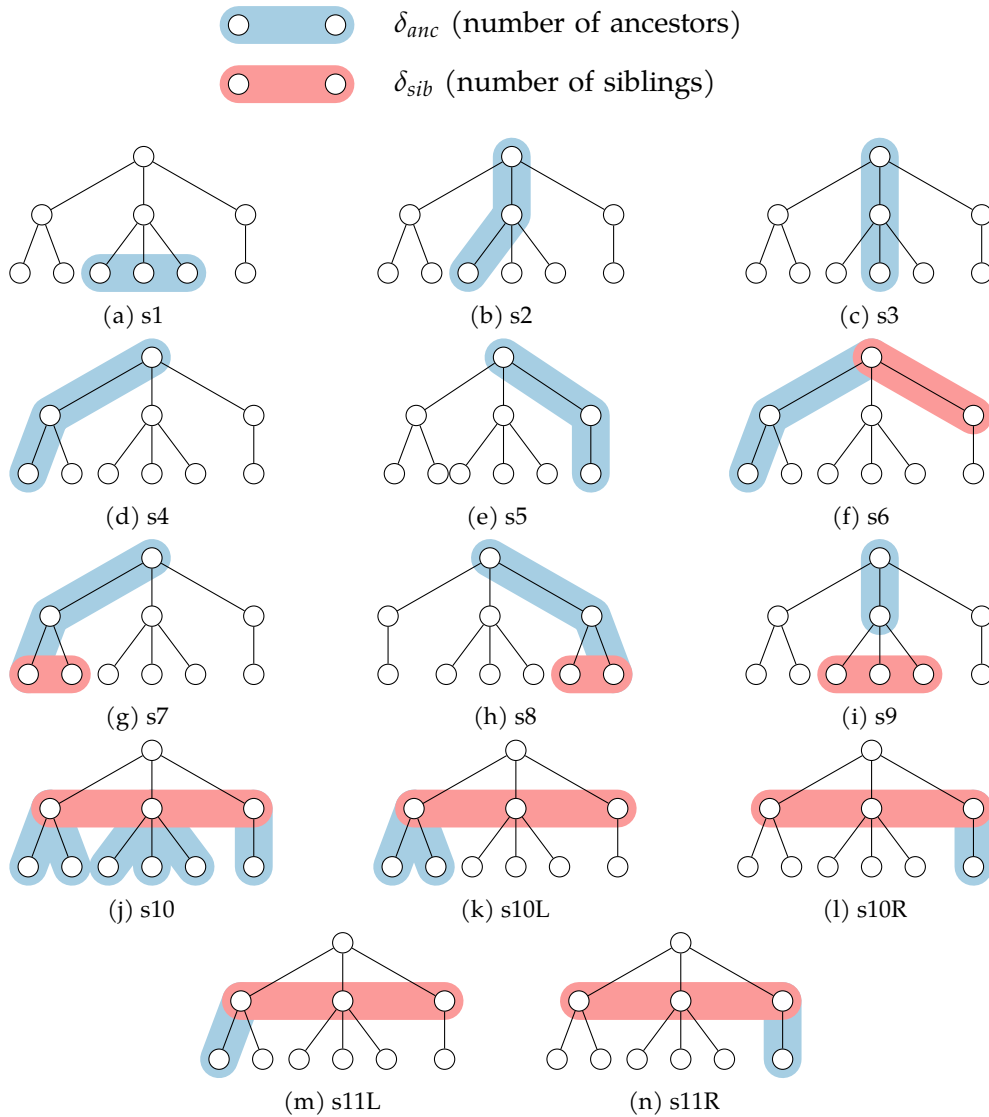


Figure A.1.: Different DT-gram shapes tested in preliminary experiments. Based on research by Pobitzer [71]. δ_{anc} and δ_{sib} are the size of the dimensions of the substructures marked in that color. For visibility reasons, wildcard nodes are not displayed in these diagrams.

Reddit Comments

B.1. JSON structure of Reddit comment

Example of a Reddit comment in json format as contained in the original data dump:

```
1 {
2   "archived": false,
3   "author": "TistedLogic",
4   "author_created_utc": 1312615878,
5   "author_flair_background_color": null,
6   "author_flair_css_class": null,
7   "author_flair_richtext": [],
8   "author_flair_template_id": null,
9   "author_flair_text": null,
10  "author_flair_text_color": null,
11  "author_flair_type": "text",
12  "author_fullname": "t2_5mk6v",
13  "author_patreon_flair": false,
14  "body": "Is it still r/BoneAppleTea worthy if it's the
15         opposite?",
16  "can_gild": true,
17  "can_mod_post": false,
18  "collapsed": false,
19  "collapsed_reason": null,
20  "controversiality": 0,
21  "created_utc": 1538352000,
22  "distinguished": null,
23  "edited": false,
24  "gilded": 0,
25  "gildings": {
26    "gid_1": 0,
27    "gid_2": 0,
28    "gid_3": 0
29  },
30  "id": "e6xucdd",
31  "is_submitter": false,
32  "link_id": "t3_9ka1hp",
33  "no_follow": true,
34  "parent_id": "t1_e6xu13x",
```

```

34  "permalink": "/r/Unexpected/comments/9ka1hp/
      jesus_fkng_woah/e6xucdd/",
35  "removal_reason": null,
36  "retrieved_on": 1539714091,
37  "score": 2,
38  "send_replies": true,
39  "stickied": false,
40  "subreddit": "Unexpected",
41  "subreddit_id": "t5_2w67q",
42  "subreddit_name_prefixed": "r/Unexpected",
43  "subreddit_type": "public"
44  }

```

B.2. Examples of Excluded Comments

Example of a message with vocabulary that is too small:

```

subreddit: /r/thanosdidnothingwrong
link id: t3_8vl3iz
comment id: e1ofray
TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT
TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT
TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT
TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT
TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT
TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT
TEXT TEXT TEXT TEXT

```

Example of a message that is too short after removing punctuation:

```

subreddit: /r/todayilearned
link id: t3_8vnjz8
comment id: e1pfsck
.- .....  .- - / - .....  . / ..-.  ..- -.-.  -.-
/ -..  ..  -..  / -.- - ..- / .- ..- ...  - / ..-.
..- -.-.  -.- ..  -  - / ...  .- -.- / .- -...
- ..- - / - .  -.- / -.- - ..- / ..-  ..  - - .-
..  . / ...  .....  ..  - ..-.. / ..  .-..  .-..
/ .....  .- .....  . / -.- - ..- / -.- - .- -.- / ..
/ - .  .-  .- -..  ..- .- - .  -.. / - - .-  / -
..-

```

Authorship Attribution Benchmark Datasets

C.1. Project Gutenberg Preamble Example

The original files from the project Gutenberg include a header in each file containing meta-data. In the following example, the actual novel starts on line 40, and in the preprocessing steps, all lines prepending it are removed.

```
1 The Project Gutenberg EBook of The Phantom 'Rickshaw and Other Ghost
2 Stories, by Rudyard Kipling
3
4 This eBook is for the use of anyone anywhere at no cost and with
5 almost no restrictions whatsoever. You may copy it, give it away or
6 re-use it under the terms of the Project Gutenberg License included
7 with this eBook or online at www.gutenberg.org
8
9
10 Title: The Phantom 'Rickshaw and Other Ghost Stories
11
12 Author: Rudyard Kipling
13
14 Posting Date: December 29, 2008 [EBook #2806]
15 Release Date: September, 2001
16 Last Updated: October 7, 2016
17
18 Language: English
19
20 Character set encoding: UTF-8
21
22 *** START OF THIS PROJECT GUTENBERG EBOOK THE PHANTOM 'RICKSHAW **
23
24 Produced by David Reed
25
26 THE PHANTOM 'RICKSHAW AND OTHER GHOST STORIES
27
28 By Rudyard Kipling
29
30     *       *       *       *       *
31
32     The Phantom 'Rickshaw
33     My Own True Ghost Story
34     The Strange Ride of Morrowbie Jukes
35     The Man Who Would Be King
36     "The Finest Story in The World"
37
38     *       *       *       *       *
39
40 THE PHANTOM 'RICKSHAW
41
42     May no ill dreams disturb my rest,
43     Nor Powers of Darkness me molest.
44     --_Evening Hymn._
```

Universal Grammar Features

D.1. Universal Part-of-Speech Tags

The following list contains small examples of each of the universal POS tags that are used by the software in this thesis. A full description and additional information can be found on the universal dependency homepage²⁷.

Tag	Name	Example
ADJ	adjective	The <i>green</i> car
ADP	adposition	<i>During</i> the match
ADV	adverb	You did <i>well</i>
AUX	auxiliary	When <i>will</i> you do this
CCONJ	coordinating conjunction	I passed, <i>so</i> i celebrated.
DET	determiner	<i>The</i> term is over.
INTJ	interjection	Did you do it? <i>No!</i>
NOUN	noun	The <i>term</i> is over.
NUM	numeral	I have <i>three</i> sisters.
PART	particle	It's hard <i>to</i> understand.
PRON	pronoun	I disagree with <i>you</i> .
PROPN	proper noun	He lives in <i>New York</i> .
PUNCT	punctuation	The part-of-speech (POS) tags
SCONJ	subordinating conjunction	I think <i>that</i> I will leave.
SYM	symbol	I owe him 20\$.
VERB	verb	The cat <i>meows</i> .
X	other	It sounded like <i>SCRRRTCH</i> .

D.2. Universal Dependencies


The following list contains small examples of each of the universal dependencies that are used by the software in this thesis. A full description and additional information can be found on the universal dependency homepage²⁸.

²⁷<https://universaldependencies.org/u/pos/>, accessed 2022-05-13

²⁸<https://universaldependencies.org/u/dep/>, accessed 2022-05-13

Appendix D. Universal Grammar Features

acl adnominal clause	a dog named cookie
appos appositional modifier	Rex, , died.
advcl adverbial clause	She entered the room while sad
advmod adverbial modifier	He bakes cakes happily
amod adjectival modifier	Sam eats fresh apples
appos appositional modifier	Sam, , marries.
aux auxiliary	You should leave.
case case marking	The dog 's nose
cc coordinating conjunction	Bob is lazy and rich.
ccomp clausal complement	He claims you like singing.
clf classifier	Not used in languages in this thesis
compound compound	The phone book

conj conjunct 
Bob is lazy and rich.

cop copula


csubj clausal subject 
What I read bothers me

dep unspecified dependency – fallback for parsers if they can't find a better rule.

det determiner 
The phone book

discourse discourse element 
Hm, I don't know.

dislocated dislocated elements 
I like it, the book.

expl expletive 
There is no time.

fixed fixed multiword expression 
I like her as well as him.

flat flat multiword expression 
New York City

goeswith goes with 
never the less

iobj indirect object 
She gave me a raise

Appendix D. Universal Grammar Features

list list
Good weather, nice people, tasty food.

mark marker
He claims that you sing.

nmod nominal modifier
The dog 's nose

nsubj nominal subject
I like you.

nummod numeric modifier
Sam ate three apples.

obj object
I like you.

obl oblique nominal
I was chased by the dog.

orphan orphan
I like apples and Sam bananas.


parataxis parataxis
Sam, she said, likes apples.

punct punctuation
Go home !

reparandum overridden disfluency
Turn right no left


root root of sentence²⁹

ROOT I like you.




vocative vocative

Guys, take it easy!



xcomp open clausal complement

You look amazing.



²⁹Technicall, the root is part of every parsed sentence, but it is omitted in the other exmaples for clarity.