# Master's Thesis

## in Computational Linguistics

---

# Deconstructing Constructed Languages

---

*Author:*
Connor Kirberger

*Supervisors:*
Çağrı Çöltekin
Christian Bentz

Seminar für Sprachwissenschaft
Eberhard-Karls-Universität Tübingen

December 2023

Hiermit versichere ich, dass ich die Arbeit selbständig verfasst, keine anderen als die angegebenen Hilfsmittel und Quellen benutzt, alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe und dass die Arbeit weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen ist und dass die Arbeit weder vollständig noch in wesentlichen Teilen bereits veröffentlicht wurde sowie dass das in Dateiform eingereichte Exemplar mit den eingereichten gebundenen Exemplaren übereinstimmt.

I hereby declare that this paper is the result of my own independent scholarly work. I have acknowledged all the other authors' ideas and referenced direct quotations from their work (in the form of books, articles, essays, dissertations, and on the internet). No material other than that listed has been used.

Tübingen, September 28, 2024

_____
Firstname Surname

# Contents

# 8 Appendices 41

## Abstract

Write the abstract here.

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| **API** | Application Programming Interface |
| **NLP** | Natural Language Processing |
| **PCA** | Principal Component Analysis |
| **TF-IDF** | Term Frequency - Inverse Document Frequency |
| **RNN** | Recurrent Neural Network |
| **LSTM** | Long Short-Term Memory |
| **SVM** | Support Vector Machine |
| **XML** | eXtensible Markup Language |
| **TTR** | Type-Token Ratio |
| **MATTR** | Moving-Average Type-Token Ratio |
| **MTLD** | Measurement of Textual Lexical Diversity |
| **IAL** | International Auxiliary Language |
| **SVO** | Subject-Verb-Object |
| **SOV** | Subject-Object-Verb |
| **IALA** | International Auxiliary Language Association |
| **LFN** | Lingua Franca Nova |
| **CSV** | Comma-Separated Values |
| **MAP** | Maximum a Posteriori |
| **MSE** | Mean Squared Error |
| **MDI** | Mean Decrease in Impurity |
| **MDA** | Mean Decrease in Accuracy |
| **AUC** | Area Under the Curve |
| **ROC** | Receiving Operating Characteristic |
| **PRC** | Precision-Recall Curve |

# 1 Introduction & Motivation

Constructed languages—also called artificial languages, invented languages, planned languages, engineering languages, glossopoeia, or more simply as "conlangs" (Ball 2015)—are languages that are consciously and purposefully created for some intended use, usually being defined in antithesis to the spontaneous and organic method in which natural languages arise and develop (Sanders 2016). These variations of the term are often, but not always, used interchangeably, as linguists do not all agree upon a core term due to personal preferences (Adelman 2014), and there are sometimes differences in nuance depending on the context in which they appear. This thesis will mainly refer to them as constructed languages for simplicity.

The intended uses for which they are created can range broadly. Some are made specifically for fictional media, often seen in the genres of fantasy or science-fiction, with some more well-known examples being J. R. R. Tolkien's Elvish languages (e.g., Quenya, Sindarin, Nandorin) found in the world of Middle-earth in his writings, Marc Okrand's Klingon language from the Star Trek universe, and David J. Peter's Dothraki language used in George R. R. Martin's A Song of Ice and Fire novels along with their television adaptation, Game of Thrones (Jeffrey Punske (editor) 2020). Others are created to function as international auxiliary languages (IALs)—languages planned for the use of international and cross-cultural communication (Gobbo 2016). The most well-known example (based on estimated number of speakers) of these is Esperanto, created in the 19$^{\text{th}}$ Century by L. L. Zamenhof. Typically, constructed languages are distinguished and categorized based on these communicative functions. This will be discussed more comprehensively in section 2.

Despite being defined in contrast to one another, however, constructed and natural languages are not necessarily opposite to one another characteristically. Aside from their origins, the boundaries between the two are not always clear when analyzed in greater detail (Goodall 2022). For example, K. Schubert (1989) argues that some languages which are considered "natural" have some degree of artificiality, such as standardized written German and English differing from their spoken forms, and that the reverse is also true of some languages which are considered "artificial" because they draw from aspects of natural languages. As such, he believes human languages exist on a continuum of the two labels, rather than in the binary distinction—a view echoed by other linguists as well (Novikov 2022).

In many ways, the investigation into the disparity between the these two kinds of languages overlaps with the broader debate regarding what constitutes a language. Central to this debate is the search for linguistic universals—

properties shared by all languages (Mairal and Gil 2006). The concept of universals in language is recognized as one of the most important areas of research in linguistics (Christiansen, Collins, and Edelman 2009) and has served as a foundation for much linguistic theory, especially in more recent history, stemming largely from the influential theories and works of Greenberg (Greenberg 1970) and Chomsky (Chomsky 1957; Cook and Newson 2007).

Analyzing their surface structures can reveal whether or not constructed languages adhere to the same linguistic conventions as their natural counterparts. If machine learning models fail to successfully distinguish between the two, it may reinforce the notion that these universals are present in all languages, regardless of origin. Conversely, the models succeeding may suggest the opposite. In short, the primary motivation behind this thesis is to contribute to this ongoing debate through the application of machine learning, and a desire to learn more about the fascinating genre of constructed languages.

## 1.1   Scope of Study & Research Question

The present work analyzes various linguistic features and seeks to successfully discriminate a language as being either natural or constructed based on these. More specifically, the scope of this study includes both binary classification and anomaly detection, with the models being trained on a set of selected features rather than raw text data.

Because of the wide-ranging nature of conducting such a broad analysis, there are of course many features left unconsidered or excluded, intentionally or otherwise. With this in mind and following the precedent set by other related research on this topic, the main focus for linguistic features relate to entropy, morphological complexity, and overall lexical diversity.

The following is a breakdown of the structure of this thesis from here onward: the next section provides relevant background information, including an overview on constructed languages and a comprehensive review of related literature that examines the prior theoretical groundwork laid for exploring linguistic similarities and differences between constructed and natural languages; section 3 covers in detail the methodology taken in this research, from an explanation of the data used to the various experiments performed; section 4 presents the results of the study and discussion of these follows in section 5; lastly, section 6 consists of a conclusion as well as elaboration for possible future work.

# 2 Background

The vast landscape of linguistic research comprises a myriad of literature delving into the intricacies of languages, both natural and constructed. As this paper is concerned with constructed languages in particular and possible distinctive properties they may have, this section begins with a brief overview of their history and development, which provides some relevant context. Following this is an overview of some related literature, which is relevant to understanding the motivation behind the various computational approaches I employ in my experiments.

## 2.1 History of Constructed Languages

Okrent (2009) states, "The history of invented languages is, for the most part, a history of failure." She may be justified in saying this, depending on one's definition of failure in this context. From past to present, the total number of constructed languages may be as high as a thousand (Libert 2016; K. Schubert 1989; K. Schubert et al. 2001), with hundreds proposed for the purpose of being IALs in Europe alone (K. Schubert et al. 2001). Yet of these, only Esperanto is commonly considered to be successful in achieving its creator's intended goal of world-wide use as an auxiliary language (or rather that it is by far the most successful), with very few others even coming close, having a conservative estimation of two million speakers (Okrent 2009).

While the construction of languages is possibly as old as human history, they typically were not written down and were limited to in-group communication (Gobbo 2016). The first documented endeavors came out of religious contexts and were likely used as secret languages, intentionally obscured and incomprehensible to lay people. In the 12th century, abbess Hildegard of Bingen described and recorded a lexicon for Lingua Ignota, a Latin name meaning "unknown language". While extensive documentation of it (i.e., a grammar) was never found, it possessed a semiotic system based on Latin, German, and Greek. Later in the 14th century, a group of Sufi mystics created Balaibalan, a language written in the Ottoman Turkish alphabet and which incorporated features of Persian, Turkish, and Arabic languages (Novikov 2022).

Interest in creating such languages picked up in the 17th century with the rise of so-called philosophical languages. In contrast to the last two, these languages were made to be more precise, less ambiguous, and better allow for philosophical reasoning (compared to natural language), such as by organizing world knowledge into hierarchies (Goodall 2022). Notable figures involved in making these include Francis Lodwick, Gottfried Leibniz, and John Wilkins, the latter of whose being arguably the most well-known and influ-

(1)      special > creature > distributively > substances > animate > species > sensitive >
         sanguineous > beasts > viviparous > clawed > rapacious > oblong-headed > European >
         terrestrial > big > docile

Figure 2.1: Wilson's expression of "dog" in his philosophical language (Goodall 2022)

ential.   Wilkins created a system of semantic categorization, cataloging all concepts in the universe (Okrent 2009), and then published his proposed language (Wilkins 1968).   An example of this hierarchal categorization can be seen in Figure 2.1.

In the 19ᵗʰ and 20ᵗʰ centuries the focus for language construction, especially in Europe, shifted to that of making international auxiliary languages (IALs) intended to better enable communication across language barriers, i.e., people who do not share a similar language (Goodall 2022).   Notably, this means they were generally (though not always) designed to resemble natural language, with choice exceptions being the simplification of certain linguistic features.   The surge in need for IALs correlated with the increase in prevalence and accessibility regarding international travel and communication at the time. Such languages were also described as "neutral" (Large 1985), in the sense that individual advantages amongst speakers and learners would, theoretically, not exist due to IALs being second languages to everyone (Gobbo 2016).   That being said, many of the most prominent examples (e.g., Volapük, Interlingua, Esperanto, Ido) are derived from the Indo-European language family (Novikov 2022; Goodall 2022), so such a description might not be apt.   Overall, IALs can be viewed as an intended rival to natural languages, which is one reason why all of the constructed languages analyzed in the present work are IALs. A more detailed explanation of each is provided in Section 3.1

Lastly, there exist constructed languages that have been made for experimental, artistic, literary, or fictional purposes.   In contrast to IALs, these languages are not made with the intention of replacing existing languages for everyday communication.   Instead, their creators want to push the boundaries of language, test scientific hypotheses like linguistic relativity, or create a world, as is the case for the fictional examples provided in Section 1. Some other examples in this category include Solresol, a language that uses musical notes; Láadan, a language designed to be inherently feminist (i.e. more capable of expressing the female experience); and Loglan, a self-described "logical" language whose morphology and syntax are based on predicate logic (Adelman 2014).   Though it would be inaccurate to describe such languages as being only a recent invention, popularity in their conceptualization largely grew in the later part of the 20ᵗʰ century.

While all share the defining characteristic of having been purposefully cre-

ated, the linguistic features of constructed languages (e.g., phonetic, morphological, syntactical, lexical, orthographic) can vary immensely depending on factors such as, for example, their intended purpose for use or the other languages they draw from. An example of this was observed by Gobbo (2016) in secret languages, specifically their tendency to have more complicated features, such as morphological irregularities, "in order to preserve their secrecy." Contrast to this are IALs, which have the opposite tendency for the sake of ease of communication and second-language acquisition, reflected in commonly assigned features such as SVO word orders, head-initial relative clauses, fronted *wh*-phrases, and morphological regularity (Goodall 2022; Gobbo 2016). Section 2.2 further examines research focused on linguistic features of these languages.

In addition to this classification based on their intended communicative functions, i.e. as philosophical or international auxiliary languages, there are also taxonomies based on other criterion. For example, another frequently used distinction is that of *a priori* and *a posteriori* (Schreyer and Adger 2021; Gobbo 2008; K. Schubert 1989; K. Schubert et al. 2001; Novikov 2022; Adelman 2014; Tonkin 2015). Languages described as being *a priori* are structurally entirely new (Tonkin 2015) and not based on existing languages, whereas so-called *a posteriori* languages are the opposite, drawing from aspects of specific natural languages (Schreyer and Adger 2021). Gobbo (2008) also proposed the dichotomy of *exoteric* (secret) and *esoteric* (public) languages, derived from Bausani (1974). Similar to critiques regarding the distinction between constructed and natural, such dichotomies for categorizing constructed languages are also argued by some linguists to be more accurately described as scales instead, with many languages falling somewhere in the middle (Novikov 2022). A final noteworthy classification scheme often cited by other linguists comes from BLANKE (1989) in the form of three classes: project, semi-planned, and planned. In short, these correspond to a set of steps that a constructed language must go through before it can be considered a "real" language (K. Schubert et al. 2001).

A two-dimensional taxonomy for constructed languages containing several notables examples is shown in Figure 2.2 (Gobbo 2016).

## 2.2    Prior Studies

In contrast to the abundance in literature and cross-linguistic analyses done on natural languages, similar research which also includes constructed languages is relatively sparse. In particular, while there is research that analyzes specific instances of linguistic differences between certain natural and constructed languages, large-scale cross-linguistic studies which utilize computational meth-
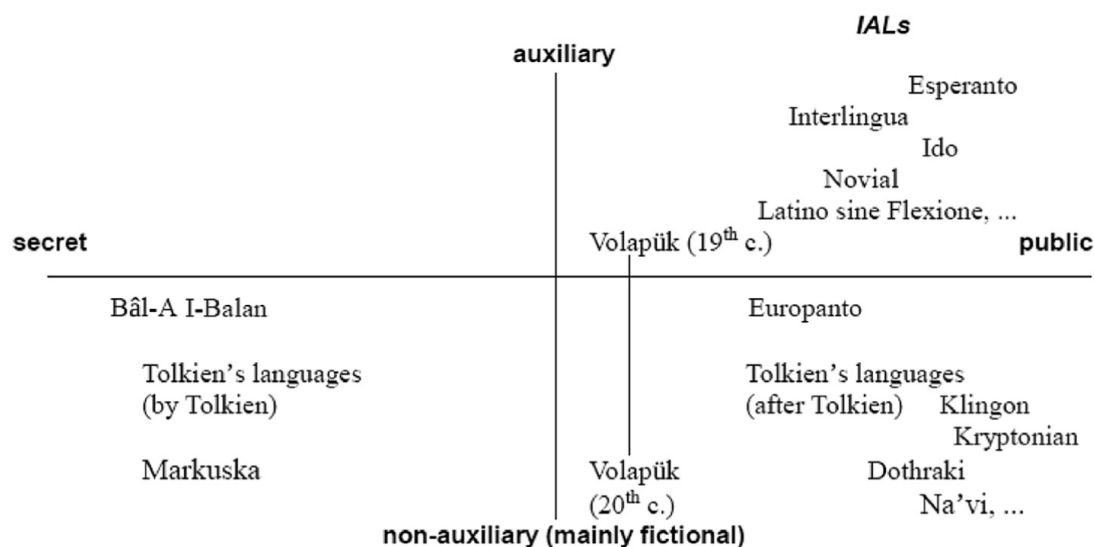
Figure 2.2: A taxonomy of constructed languages (Gobbo 2016)

ods to classify the two based on linguistic features are practically nonexistent. Consequently, the present study is a somewhat novel approach. However, there is precedent for this research and the specific features examined, as well as computational approaches used, which this section will describe.

As noted in the previous section, the creation of IALs often involved the intentional simplification of particular linguistic features to facilitate language acquisition, for instance having more regularity in their morphological systems. Intuitively, then, one would assume this translates to measurable differences in various aspects of linguistic complexity when compared to natural languages, which often have irregularities as a result of their development and evolution. When comparing Volapük and English, **GObbo2016** concluded that

Much of the literature on constructed languages focuses on Esperanto specifically.

Another feature examined is entropy. Originating from information science, entropy was introduced by Shannon (1949) as a measurement of uncertainty or surprisal for an event, with high surprisal being inversely proportionate to the amount of information conveyed by the event's occurrence. In NLP,

# 3    Methodology

In this section, I introduce the dataset for this paper and discuss the steps taken for preprocessing it, followed by discussing in detail the features examined along with the different methods involved in extracting them from the data, and finally the classifiers employed on the feature set. A brief description of the various APIs and libraries used is also included in 3.3.

Since this study involves many different experiments and elements being performed and analyzed, I will begin by explaining an overview of what all was done. The number of possible features and measurements of linguistic complexity which could be analyzed in such a study is extensive to say the least; however, the scope of this thesis focuses mainly on empirical measurements relating to lexical diversity, morphological complexity, and entropy, along with some others measurements which are commonplace to calculating linguistic complexity, and thus seemed appropriate to also include. More specifically, the features investigated are average word length, average sentence length, type-token ratio (TTR) of morphemes, average number of segmentations in a word, average number of forms per lemma, lexical TTR, moving-average type-token ratio (MATTR), measurement of textual lexical diversity (MTLD), lexical entropy, text entropy, and character and word distribution entropies. Once the values of these were calculated for each language, the task became that of supervised classification and unsupervised anomaly detection with five machine learning models: a one-class support vector machine (SVM), local outlier factor, random forest, isolation forest, and decision tree. Lastly, the methods for evaluating the performances of these models are discussed, as is the Principal Component Analysis (PCA) performed on the data for dimensionality reduction.

## 3.1    Data

In total, twenty-four languages are analyzed in this study. Six of these are constructed languages: Esperanto, Ido, Interlingua, Lingua Franca Nova, Volapük, and Kotava. The remaining eighteen are natural: German, English, Spanish, Polish, Vietnamese, Indonesian, Turkish, Tagalog, Hungarian, French, Finnish, Italian, Dutch, Occitan, Danish, Swedish, Afrikaans, and Icelandic.

For consistency, only languages which are written using the Latin alphabet (including the use of diacritics) were chosen. This is mainly because the constructed languages in the dataset all use Latin alphabets, so the selection of natural languages followed the same criteria. Moreover, it allows for more uniform cross-linguistic analysis of features which may be sensitive (e.g. in the case of character entropy) to writing systems.

### 3.1.1 Constructed Languages in the Dataset

All of the constructed languages in the dataset are IALs, with most of them resembling natural (particularly various European) languages. I will briefly introduce each of them in this section, explaining where they come from, some notable typological features they have, and how they compare to both one another other and their natural counterparts.

Esperanto, the most widely-spoken constructed language and considered by many to be the most successful (Gobbo 2008), was created in 1887 by Polish ophthalmologist L. L. Zamenhof. Zamenhof's goal was to create a neutral, easy-to-learn language that would facilitate international communication. Esperanto is a highly regular language, with consistent grammar and a simplified, phonetic spelling system. It draws its lexical roots and syntax primarily from Romance, Germanic, and Slavic languages (Gobbo 2008; Gobbo 2011), making it recognizable and familiar to speakers of many European languages, while also intentionally being made to have a comparatively simpler grammar that avoids some complexities found in natural languages, such as irregular verbs or noun cases. It also has a strong global community with speakers around the world, an array of written literature, and even a number of native speakers who learn it from birth—a distinguishing trait which sets it apart from other constructed languages (Goodall 2022). As a result of its success, Esperanto also serves as a direct influence for many other constructed languages that have come after it, one being Ido.

Ido is a reform of Esperanto that was proposed in 1907 by a group of linguists led by Louis Couturat, a French philosopher and mathematician, and in fact is an Esperanto word meaning "offspring" (K. Schubert et al. 2001). Its creators sought to address what they saw as imperfections in Esperanto, particularly those related to orthography and morphology. For instance, Ido avoids the use of the accusative case and reforms some Esperanto words to make them more universally recognizable. Overall, though, Ido still retains much of Esperanto's vocabulary and basic structure, and the two are mutually intelligible to a large extend (Goodall 2022; K. Schubert et al. 2001). Like most of the remaining constructed languages to be discussed in this section— with the exception of Volapük—Ido has small but a dedicated community of speakers and enthusiasts.

Interlingua was developed by the International Auxiliary Language Association (IALA) with the assistance of linguist Alexander Gode, officially being published in 1951. The idea behind Interlingua for it to most recognizable to the greatest number of people without requiring prior study (Goodall 2022), with most attention having been spent on its lexicon. The IALA's stated goal was to not so much create a new international language, but rather present a standardized international vocabulary (Large 1985) ("international" here

basically referring to Western Europe). It is largely derived from and resembles Romance languages (with lesser influence from Greek and Germanic languages) (K. Schubert et al. 2001). In fact, this intentional resemblance extends even to morphological irregularities such as allomorphy, with other irregularities also being introduced to the language to make it appear more natural (Goodall 2022), a contrast to other IALs like Esperanto.

Volapük was created in 1879 by Johann Martin Schleyer, a German Catholic priest who believed the language had been given to him by God. It features highly agglutinative structure and regular, yet complex, morphology. While being derived mainly from English, German, and Latin, roots in Volapük differ significantly to the point of being unrecognizable to speakers of these languages (Goodall 2022). Despite being argued to be the first successful constructed language due to its rise in popularity, having amassed a large number of supporters worldwide along with the formation of clubs and societies (Gobbo 2016), various issues regarding its complexity led to a rapid decline and eventual fall from usage in favor of Esperanto.

Lingua Franca Nova, also abbreviated as LFN, is a relatively recent constructed language created by linguist C. George Boeree in 1998. Its lexicon is based mainly on Romance languages, specifically French, Italian, Portuguese, Spanish, and Catalan, while its grammar is based on Romance creole languages (Pawlas and Paradowski 2020). In particular, inspiration came from the similarly-named Mediterranean Lingua Franca, a pidgin that developed for trade in the Mediterranean basin and was used from the $11^{\text{th}}$ to $18^{\text{th}}$ centuries, as well as from other creoles, such as Haitian Creole. It can be written in both Latin and Cyrillic scripts, though this dataset only contains the former.

The last constructed language used is Kotava. Created by Staren Fetcey in 1978, Kotava stands out in this dataset as being an attempt at creating a culturally neutral *a priori* language, free from any biases or influences of existing languages and based on a philosophy of linguistic egalitarianism. This intentionally designed uniqueness is apparent in several of its linguistic systems, from morphology to syntax. For example, though word order in Kotava is not imposed, the most frequently used one is OSV, which is exceedingly rare in natural languages in contrast. Other unique features include a $4^{\text{th}}$ person plural, object complements being introduced by a transitive preposition, and a lack of declension (Fetcey and Comité Linguistique Kotava 2013).

Finally, it is worth drawing attention to the fact that each of these languages were created based on various European languages, with the exception of Kotava. Consequently, this may influence the models used in the experiments and be visible in the results. This will be explored in greater detail later in Section 5.

Table 3.1 shows the dataset's constructed languages together with the main

| Conlang | Source Languages/Families |
|---|---|
| Esperanto | Romance, Germanic, Slavic |
| Interlingua | Romance |
| Lingua Franca Nova | Romance |
| Volapük | Germanic |
| Kotava | N/A |
| Ido | Romance, Germanic, Slavic |

Table 3.1: Constructed languages used in the study, together with their main respective source languages from which they were designed.

source languages they draw from by design (with *N/A* for Kotava meaning *not applicable*). Note, however, that this is not an exhaustive list of all of their language influences.

## 3.1.2 Natural Languages in the Dataset

The natural languages included in this study represent a broad spectrum of linguistic diversity, comprising a variety of families, geographic regions, and typological features. Although this representation is not necessarily equal in distribution, it is meant to serve as a contrast to the constructed languages in the dataset, which lack a similar extent of variety due to overwhelmingly having the same source languages. However, rather than delving into the same level of details for each of the eighteen languages here as I did for the constructed ones, I will instead introduce them by focusing mostly on their collective significance and summarizing some of their relevant typological traits.

In total, five major language families are represented. The largest of these—based on number of speakers worldwide as well as the number of languages in the dataset (twelve)—is Indo-European, with several of its branches being included. English, German, Dutch, Afrikaans, Swedish, Icelandic, and Danish all belong to the Germanic branch. Similarly, Italian, Spanish, French, and Occitan are part of the Romance branch, all being descendants of Latin. Polish is an outlier as the only represented language from the Slavic branch.

The other four families span less representation in the dataset in comparison, but were nevertheless included so as to have more variety in linguistic features. These include the Uralic languages, consisting of Hungarian and Finnish, which are the most widely-spoken and thus representative of their group, as well as Austronesian (Tagalog and Indonesian), Austroasiatic (Vietnamese), and Turkic (Turkish).

One notable typological feature of all of these languages which relates to the scope of this study is that of their morphology. Traditionally, classification of morphological systems follows two paradigmatic axes: inflection versus derivation and agglutination versus flexion.

### 3.1.3   Wikimedia

The data for this thesis comes from Wikimedia dump files. Wikimedia is a global movement and community founded on shared values, whose goal is to provide free and openly accessible information to everyone in the form of massive collaborative projects (which include, among others, the widely-used Wikipedia and Wiktionary). For a large, cross-linguistic study, massive databases with open-access make for an ideal source for corpora. Most importantly, the projects are multilingual, meaning data is available in a considerable number of different languages—including several constructed ones. This allows for composing a set of corpora which is adequately parallel to each other and from the same domain. Additional constructed languages which are also available from the dumps—but were not included in the present study due to having a much smaller amount of data—are Novial, Interlingue, and Lojban.

The dump files provide detailed, archived snapshots of the content from Wiki repositories for a specified point in time and are available in different formats. All dumps used were XML-formatted and from the 2024-07-01 archive, containing articles together with their metadata[1]. It is also worth mentioning here that there are some drawbacks to using these dumps for the present study. The files sizes vary considerably depending on the language, with the largest being roughly 22 gigabytes (English) and the smallest around 4 megabytes (Lingua Franca Nova), meaning all files do not contain the exact same articles. Additionally, the open and collaborative nature of Wikimedia means the articles are often authored by a multitude of different people, which can result in inconsistencies in the texts, such as with writing style. Similarly, it may also produce an imbalance in the amount of information provided across languages, with the same article in one language being considerably more detailed than in another, and inconsistent or low-quality translations, as Novikov (2022) noted to be the case for Wikipedia articles in Volapük. Thus, while Wikimedia was decided as the best available option for the task at hand, there are some unfavorable aspects of using it which may influence the results; this will be discussed more in Section 5.

## 3.2   Data Preprocessing

Preprocessing text data is essential for natural language processing (NLP) tasks, so meticulous effort was made to thoroughly clean all of the texts and obtain as close to a set of parallel data as possible.

Text data was first extracted from the Wikimedia XML-formatted dump files with the use of WikiExtractor[2], a Python script (Attardi 2015) that I

---

[1]`https://dumps.wikimedia.org/backup-index.html`
[2]GitHub repository for WikiExtractor: `https://github.com/attardi/wikiextractor`

adapted by adding a limit to the number of articles in order to make extraction of the largest of the files (English in particular) less demanding and quicker. The output is a simple text file, which is a much easier format to clean and work with.

I then used several regular expressions to remove general, unnecessary text from each file such as page titles, section headers, links, fragments, HTML tags, braces, and all other non-alphabet symbols aside from periods. This also includes the removal of parantheses and their contents. The text was then made all lowercase and split by the periods—while also attempting to account for abbreviations—to make separate sentences. This was done mainly to enable more accurate measurement of entropy later.

Following this, foreign symbols (i.e., characters not part of a particular language's alphabet) were removed for each text/language, as occasionally proper nouns, loanwords, etc. would appear in the text, which would also affect measurements of entropy, in addition to morphological segmentation and analysis. To give an example of this, there is no letter 'h' in Kotava, but this would sometimes be used in proper nouns such as 'Hiroshima'. After the text is cleaned, the remaining word left behind is 'irosima'.

Finally, each text file was truncated according to the file size of the smallest corpus, LFN, so as to have similar lengths. This was calculated based on number of words, with the limit being 630000 (since this is roughly the number of words remaining in the LFN text file after cleaning), and while preserving complete sentences. Sentences containing only one word were also removed. The end result of pre-processing was a single text file for each language, with every line in the file being a single sentence. The corpora with the smallest and largest number of words are Kotava and Danish/Volapük at 617400 and 629999 words, respectively. For number of sentences, the smallest and largest corpora are Vietnamese and Volapük at 21115 and 55920 sentences, respectively. For a full breakdown of these size for each language's text after pre-processing, refer to Table 8.1.

## 3.3   Libraries and APIs

Several libraries and APIs were used in both the feature engineering and classification steps of the experiment, and the most important of these will be briefly introduced here. In the field of machine learning, two of the most popular model frameworks used are PyTorch and TensorFlow, which are interacted with via the torch and Keras APIs, respectively. Aside from some relatively minor differences (e.g., the syntax of their code and performance optimization), they share a lot of similarities and are typically used according to personal preference. In the context of this paper, these frame-

works are used for calculating some of the entropy values from the corpora. Other machine learning models used were for morphological segmentation via MORFESSOR 2.0, a family of unsupervised, generative probabilistic models (Virpioja et al. 2013).

In addition to libraries used for constructing the model architectures, there are also ones used for the data itself. Arguably the most fundamental for this is NUMPY, which stands for Numerical Python and is used to accomplish extremely fast and efficient computation of arrays. Other essential libraries include PANDAS, used for data manipulation and analysis, and MATPLOTLIB, used for visualizing data and plotting model results. Lastly, SCIKIT-LEARN provides a wide range of tools for machine learning algorithms, data preprocessing, and model evaluation—as well as computation, thanks to it being built on top of NUMPY. The models I used for classification and anomaly detection (i.e., One-Class SVM, Isolation Forest, Local Outlier Factor, Random Forest, Decision Tree) as well as PCA come from this library. Altogether, these libraries are often used in tandem in NLP tasks due to integrating so well with one another.

## 3.4   Feature Engineering

Before classification or anomaly detection can be done with the data, an initial step of feature engineering is performed. Put simply, feature engineering is the process of transforming raw text data into a more structured and comprehensible format for machine learning models through the specific selection of its most informative and relevant features, typically through some empirical measurements, thereby increasing the model's effectiveness.

The exact methods and measurements involved in this process depend on, among other factors, the task and data. For the scope of this paper, some were only simple calculations, the simplest of these being the measurements for text complexity: the average word length and average sentence length of each text. For the rest of the linguistic features (i.e. lexical diversity, morphological complexity, entropy) analyzed, however, this process involved the use of more sophisticated algorithms and various machine learning models to derive measurements from.

The complete set of the analyzed features for each language and their computed values, rounded to 3 decimals, is shown in section 4 in Table 4.1.

## 3.4.1   Lexical Diversity

A common way of measuring the lexical diversity of a text is with TTR, with a high value indicating that a given text contains a large amount of lexical

variation. This is calculated using the formula:

$$\text{TTR} = \frac{|V|}{|N|}$$

where $|V|$ denotes the vocabulary size as the number of unique words, or types, and $|N|$ denotes the text length as the total number of words, or tokens. I then multiply this by 100 to get a percentage.

A big issue with TTR, though, is that it does not always provide an accurate assessment due to its sensitivity to text length; the longer a particular text, the higher the likelihood of repetition in words occurring, consequently affecting the calculation. To remedy this, I also calculate the MATTR, a variation of TTR proposed by Covington and McFall (2010) that uses a sliding window of a fixed-length over the text and calculates the TTR at each length of the window, which is then averaged together. This is denoted by the formula:

$$\text{MATTR}_i = \frac{1}{N - i + 1} \sum_{j=1}^{N-i+1} \frac{|\text{Types}_{j,j+i-1}|}{|\text{Tokens}_{j,j+i-1}|}$$

where $N$ is the total number of tokens in the text, $i$ is the window size, $|\text{Types}_{j,j+i-1}|$ is the number of unique words (types) in the window, and $|\text{Tokens}_{j,j+i-1}|$ is the total number of words (tokens) in the window.

While resistant to variation in overall text length, calculations for MATTR do vary based on the window size, and deciding which value to use depends on the task. Here, a length of $i = 100$ tokens was used.

Another alternative to the standard measurement of TTR is the Measurement of Textual Lexical Diversity (MTLD), which calculates the average length of sequential word strings that maintains a TTR above a specified threshold (Bestgen 2024). This was done by incrementally adding the words of a given text to a sequence and calculating the TTR at each increment. Each time the TTR fell below the threshold (here, the standard threshold of 0.720 (McCarthy and Jarvis 2010; Fergadiotis, Wright, and West 2013) was used), a counter—called a factor count—was increased by 1, and both the TTR evaluations and sequence were reset. This continued until the end of the text is reached, after which the text's total number of words is divided by the total factor count to get a value for MTLD. Then the text was reversed and the same process was repeated. The two resulting values were averaged to get the final MTLD.

Typically, there are remaining words at the end of a text that are referred to as partial factors, due to not making a full one. To still include these partial factors in the overall calculation, the TTR of the remaining words was divided by 0.280 (TTR threshold subtracted from the TTR upper-bound of 1), and the result was added to the factor count.

## 3.4.2   Morphological Complexity

Morphological complexity was analyzed as three features: morpheme TTR, the average number of segments per word, and the average number of forms per stem. To calculate these, the text data of a given corpus was first split and restructured into a list of words to then be fed to corresponding model for segmentation.

Baseline models, one for each corpus, were initiated using default parameters (`forcesplit_list=None, corpusweight=None, use_skips=False, nosplit_re=None`). The word lists were then loaded into the models, again with default parameters (`freqthreshold=1, init_rand_split=None`), with the exception of `count_modifier`, which was set to a specified logarithmic equation, expressed as:

$$\lfloor \log_2(x + 1) \rceil$$

Where $x$ is the raw frequency count of the compound (i.e. the word being segmented), and the surrounding $\lfloor$ and $\rceil$ brackets denote rounding to the nearest integer. This equation was used in order to dampen the frequency of the count of a given compound, which in turn allows the model to generalize better rather than focusing more on the compounds that occur at a higher frequency.

Finally, the models were trained using default parameters (`algorithm='recursive', algorithm_params=(), finish_threshold=0.005, max_epochs=None`). Here, 'recursive' refers to the algorithm for splitting the compounds. To briefly explain how this works without delving too deep into the math behind these segmentation models—as doing so would be beyond the scope of this thesis—training one epoch involves trying all viable two-part segmentations for every compound in the data, recursively attempting further segmentation based on the lowest cost (derived using maximum a posteriori (MAP) estimation) yielded each time and stopping once this cost falls below a given threshold (Smit et al. 2014).

Once all of the texts' compounds were segmented, the aforementioned features were computed. As an approximation, the stem of the compound was assumed to be the largest resulting segment that appeared first (in order of left to right). Remaining segments, if there were any, were thus considered morphemes. Morpheme TTR was then calculated using the same formula as lexical TTR, and the result was again multiplied by 100. Computing the average number of segments per word was simply each compound's number of segmentations summed together and then divided by the total number of compounds. Lastly, the average number of forms per stem was derived from the total number of morphemes for each unique stem (adding 1 to account for the stem itself) divided by total number of unique stems.

The preference for using mostly default parameters for the functions and models here, as well as the method for identifying the stem of a segmented compound, was to ensure uniformity, so that the results would be comparable for cross-linguistic analysis. The implications of this and possible alternative approaches will be further discussed in section 5.

### 3.4.3   Entropy

In total, five values of entropy were measured: text, lexical, reversed lexical, character distribution, and word distribution. The latter two were calculated without the use of a model, simply with the formula for Shannon's entropy for a given text:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

Where $X$ refers to the random variable (e.g. the word or character distribution of a text), $n$ is the number of distinct values (i.e. types) in the distribution, $p(x_i)$ shows the probability $p$ of each type $x_i$ occurring and is calculated by its relative frequency, and $-\log_2 p(x_i)$ represents the self-information for each type. Put together, $-p(x_i) \log_2 p(x_i)$ represents the mathematical expectation for a given type, and the sum of all of these is the entropy.

Text entropy was calculated using a character-level Long Short-Term Memory (LSTM) model built for text generation.[3] LSTMs are a type of Recurrent Neural Network (RNN) that are typically better at handling long-term dependencies in sequential data due to their gating mechanisms for retaining only the information deemed useful, consequently making them better suited for generative tasks (Dhandapani et al. 2023).

Using this model required first creating a dictionary of the input text's characters, utilizing said dictionary to map the characters in each text to their corresponding integers, then one-hot encoding them (converting them into binary vector representations). Once the text data was encoded, it was split into mini-batches to be used for training the model. A mini-batch is essentially a sliding window in the shape of an $N * M$ array of characters, where $N$ is the number of sequences (equivalent to a batch size) and $M$ is the number of steps, or length of the window. Additionally, this means the input must be divisible into full batches, so remainder text that is insufficient in size is discarded. These mini-batches are then fed to the model.

Table 3.2 shows the model's main parameters alongside their associated values as used here. All other parameters were left as the default values. The architecture comprises 2 hidden layers (`n_layers`), a dropout layer, and a fully-

---

[3]Model   adapted   from   `https://github.com/LeanManager/NLP-PyTorch/blob/master/Character-Level\%20LSTM\%20with\%20PyTorch.ipynb`

| LSTM For Calculating Text Entropy | |
|---|---|
| **Parameters** | **Values** |
| Number of epochs | 20 |
| Number of sequences (`n_seqs`) | 128 |
| Number of steps (`n_steps`) | 100 |
| Number of hidden layers (`n_layers`) | 2 |
| Number of hidden units (`n_hidden`) | 256 |
| Learning rate | 0.001 |
| Dropout rate | 0.5 |
| Optimizer | Adam |
| Criterion | Cross-Entropy Loss |
| Fraction of data for validation | 0.1 |
| Gradient clipping | 5 |

Table 3.2: Parameters of PyTorch LSTM used to calculate text entropy

connected output layer. Both `n_layers` contain 256 hidden units (`n_hidden`), which are used to store information from the input and essentially act as the memory of the LSTM. Default values for learning rate, dropout rate, and gradient clipping were used, and training was done over 20 epochs. Minimal fine-tuning was performed overall, and primarily just for the `n_seqs` and `n_steps` hyperparameters.

Additionally, a fraction (0.1) of the initial input data was set aside to be used for validation, from which the model's perplexity—a measure related to entropy—is derived. To arrive at this measurement though, the validation loss is first evaluated using (multi-class) cross-entropy. Cross-entropy is an extension of Shannon's entropy, but measures instead the difference between a model's predicted probability distribution and the true one. More formally, this is represented as:

$$H(p, q) = -\sum_{x=i} p(x) \log_q(x)$$

Where $p$ and $q$ represent the discrete predicted and true probability distributions, respectively, and with the natural logarithm $\log_e$, as is commonplace in machine learning. As the predicted distribution gets closer to the true one, the resulting cross-entropy becomes lower. Entropy $H(q)$ is thus a lower bound of cross-entropy $H(p, q)$.

The validation loss was repeatedly assessed throughout training, and the mean of these was used to calculate the perplexity. Expressed mathematically, perplexity $PP$ is simply an exponentiation of cross-entropy and can be denoted by the equation:

$$\begin{aligned} PP &= \mathrm{e}^{H(p,q)} \\ &= \mathrm{e}^{-\sum_{x=i} p(x) \log_q(x)} \end{aligned} \tag{1}$$

Therefore, the final value arrived at is the text's perplexity. This is used to represent the text entropy instead, however, due to being more interpretable while still conveying the same information about the text.

Finally, lexical entropy—at the character-level—is a measure of uncertainty in predicting the subsequent character in a given sequence, in this case a word. Reversed lexical entropy is the same thing applied to a sequence that has been reversed, which may be an informative feature for distinguishing languages whose lexicons, for example, present more prevalence in prefix or suffix constructions.

For calculating both of these features, LSTMs were again used. Similar to before, the text data was first encoded into integers via dictionaries containing the vocabulary (alphabet) of each text. In contrast to the model for calculating text entropy, however, both of these models analyze the data as individual sequences of words. This means the additional use of beginning of sequence ('<') and end of sequence ('>') characters, and each sequence being padded to the same length for uniformity.

The only significant change in the process for computing both entropies occurred in the text encoding step; for the reversed lexical entropy, the sequence was reversed prior to padding. An example of forward-facing sequences (2) and their reversed forms (3) from the English corpus is illustrated here:

$$\text{'<misinterpreted>', '<filaments>', '<assisting>'} \qquad (2)$$

$$\text{'<deterpretnisim>', '<stnemalif>', '<gnitsissa>'} \qquad (3)$$

The architecture of both models is identical. First is an embedding layer with an `input_dim` equal to one more than the size of the vocabulary, an `output_dim` of 128, and `mask_zero` set to `True` (due to padding the sequences previously). This is followed by an LSTM layer, dropout layer, and finally the output layer with a softmax activation function. Additionally, early stopping was implemented based on the validation loss and with a `patience` of 2. Table 3.3 shows an overview of the most relevant parameters of the model, for which a small degree of fine-tuning was also done. All other parameters were the default values.

A fraction (0.2) of the data was again set aside for use as the validation set. Perplexity of both models was then calculated the same way as before: an exponentiation of the cross-entropy derived from the average validation loss, using Equation 1.

| LSTMs For Calculating Lexical & Reverse Lexical Entropy | |
|---|---|
| **Parameters** | **Values** |
| Number of epochs | 100 |
| Batch size | 32 |
| Learning rate | 0.001 |
| Dropout | 0.2 |
| Optimizer | Adam |
| Criterion | Sparse Categorical Cross-Entropy Loss |
| Fraction of data for validation | 0.2 |
| Metrics | Sparse Categorical Accuracy |

Table 3.3: Parameters of TensorFlow LSTMs used to calculate lexical and reverse lexical entropy.

### 3.4.4   Dimensionality Reduction of Features

To identify the specific linguistic features which convey the most meaningful information in the data, PCA was performed. PCA is an unsupervised method for linear dimensionality reduction that transforms data from a high-dimensional space to a lower one while preserving as much variance or essential information in the data as possible (Deng et al. 2024).

Two principal components were used. Technically, maximizing the number of informative features—according to calculations of explained variance—to preserve in this data would actually lead to using more, likely three (depending on the threshold). However, reduction to only two dimensions instead allows for easier visual interpretation, especially in the context of identifying possible outliers.

Before conducting the PCA, standardization (also called Z-score normalization) was performed on the data first, given by the formula:

$$Z = \frac{x_i - \mu}{\sigma} \tag{4}$$

where $\mu$ is the mean and $\sigma$ is the standard deviation from the mean. This process scales the data to fit a standard normal distribution, thus the resulting transformed data has a mean of 0 and standard deviation of 1.

### 3.5   Classification

After computing the complete set of features, two models were trained for supervised binary classification, predicting a language in the dataset to be either natural or constructed according to said numerical features. Feature scaling (e.g. normalization, standardization) was not performed for this, as neither kind of model is sensitive to variance in the data, meaning it would have no effect.

Fine-tuning to find optimal parameters was done using the grid search method, which constructs estimators to exhaust all possible combinations of a specified set of values and evaluates each of these estimators to find the best performing one. Leave-one-out cross-validation was used in order to prevent overfitting on the small and imbalanced dataset. This approach trains a model on all the data except for one (left out) sample, which is used as the testing set to make a prediction, and repeats this process for all samples.

Additionally, to prevent bias towards the majority class, the balanced class weights were used for both models; this adjusts the misclassification cost for each class with weights that are inversely proportionate to their frequency in the data (Han, Williamson, and Fong 2021).

## 3.5.1   Decision Tree

Decision Tree is a predictive analysis model structured as a hierarchy of nodes and connecting branches, constructed using binary splits. The optimal hyperparameters found from fine-tuning are shown in Table 3.4. For the rest of the model's parameters not shown here, the default values were used.

| Decision Tree Classifier | |
| --- | --- |
| **Parameters** | **Values** |
| criterion | gini |
| splitter | best |
| max_depth | None |
| max_features | None |
| min_samples_leaf | 1 |
| min_samples_split | 2 |

Table 3.4: Fine-tuned hyperparameters for Decision Tree.

The `criterion` parameter refers to the algorithm used to determine the optimal splits for the tree, which in this case was the Gini impurity; in short, this measures the probability of incorrect classification of a random datapoint in the data. The other parameters relate to limiting the structure of the tree in various ways (i.e., the number of nodes, splits, or layers).

## 3.5.2   Random Forest

Random Forest is an ensemble method that combines many decision tree classifiers and outputs the mode of their predictions as the final classification (Oktafiani, Hermawan, and Avianto 2024). Each tree is built using different random subsets of samples and features from the data using bagging techniques, thereby decreasing the risk of overfitting and increasing accuracy for classification (Salman and Al-Jawher 2024; Oktafiani, Hermawan, and Avianto

2024). This gives them an advantage over simple decision trees, albeit usually in the context of bigger datasets than the one used here.

The fine-tuned hyperparameters along with their respective optimal values, as determined by the estimator, are shown in Table 3.5. Default values were again used for remainder parameters not shown.

| Random Forest Classifier | |
| --- | --- |
| **Parameters** | **Values** |
| n_estimators | 15 |
| max_features | None |
| max_depth | 2 |
| min_samples_split | 5 |
| min_samples_leaf | 1 |
| criterion | gini |
| bootstrap | False |

Table 3.5: Fine-tuned hyperparameters for Random Forest.

In addition to the increase in accuracy and generalization, another advantage of using Random Forest is the ability to calculate and rank the importance of each feature from the data in their contribution to the model's predictions. This was done using two measurements: Gini importance (also called mean decrease in impurity (MDI)) and permutation importance (also called mean decrease in accuracy (MDA)).

## 3.6   Anomaly Detection

The engineered feature set was also used for unsupervised anomaly detection (also called outlier detection), a task for identifying outliers (and inliers, by extension) in a given dataset. This means the models train and predict on the same data to determine possible outliers, and it differs slightly from the task of novelty detection, which detects outliers in new, previously unseen data.

For these methods, fine-tuning was again performed using a grid search approach. A notable difference with these models from the previously discussed supervised classifiers in section 3.5, however, is the lack of use of weighted classes to balance the dataset, since anomaly detection inherently assumes an imbalanced distribution of classes.

### 3.6.1   Isolation Forest

Isolation Forest is similar to Random Forest, but with a few key differences that specialize it for the domain of anomaly detection instead. Proposed by Liu, Ting, and Zhou (2009), it is an ensemble algorithm which constructs trees, here called "isolation trees", through iterative branching and partitioning the

data with randomly selected features and random split values until the data points are isolated (Xu et al. 2023). An anomaly score is then calculated based on the average path length (i.e., from the root node until the isolated point) of all the isolation trees (Rosenhahn and Hirche 2024).

Much like the classification with Random Forest, feature scaling is not required for Isolation Forest, and doing so has no effect. Table 3.6 shows the hyperparameters that were fine-tuned along with the optimal values found for each. All others were left as their default values.

| Isolation Forest | |
| --- | --- |
| **Parameters** | **Values** |
| max_samples | 0.5 |
| max_features | 1.0 |
| contamination | 0.25 |
| n_estimators | 5 |
| bootstrap | False |

Table 3.6: Fine-tuned hyperparameters for Isolation Forest.

A notable parameter used by anomaly detection models is `contamination`, which reflects the proportion of outliers in the dataset.

## 3.6.2   One-Class SVM

A One-Class SVM is an unsupervised SVM used in the domain of anomaly detection which projects data into a higher-dimensional space and finds the hyperplane that maximally separates this data from the origin of said space (Boiar, Liebig, and E. Schubert 2022). Various kernel functions are utilized for this projection, and the resulting decision boundary encompasses inliers, with datapoints outside of it being considered outliers.

An initial step of standardization was performed on the data (Equation 4), and then the model was then fine-tuned to find the optimal parameters, shown in Table 3.7.

| One-Class SVM | |
| --- | --- |
| **Parameters** | **Values** |
| kernel | poly |
| degree | 3 |
| coef0 | 0.3 |
| gamma | 0.0001 |
| nu | 0.5 |

Table 3.7: Fine-tuned hyperparameters for One-Class SVM.

Here, `nu` is

### 3.6.3   Local Outlier Factor

Local Outlier Factor is a proximity-based approach to anomaly detection which detects outliers using the k-nearest neighbor algorithm.

Data was again scaled using standardization. Unlike the other two models for anomaly detection, however, Local Outlier Factor trained on the entire dataset and identified the outliers present. Table 3.8 shows the model's hyperparameters that were fine-tuned; the rest were left as their defaults.

| Local Outlier Factor | |
|---|---|
| **Parameters** | **Values** |
| n_neighbors | 5 |
| algorithm | auto |
| leaf_size | 30 |
| contamination | 0.25 |

Table 3.8: Fine-tuned hyperparameters for Local Outlier Factor.

## 3.7   Model Evaluation

Because of the limited size of the dataset, thorough evaluation of both the supervised binary classifiers and the unsupervised anomaly detection models using testing sets was not possible. Instead, their performance was evaluated Precision, recall, and F1-score were evaluated.

# 4 Results

This section reports the results of the methods implemented, and the measurements derived from this, for feature engineering. Following this are the results of the various models used for both the classification and anomaly detection tasks.
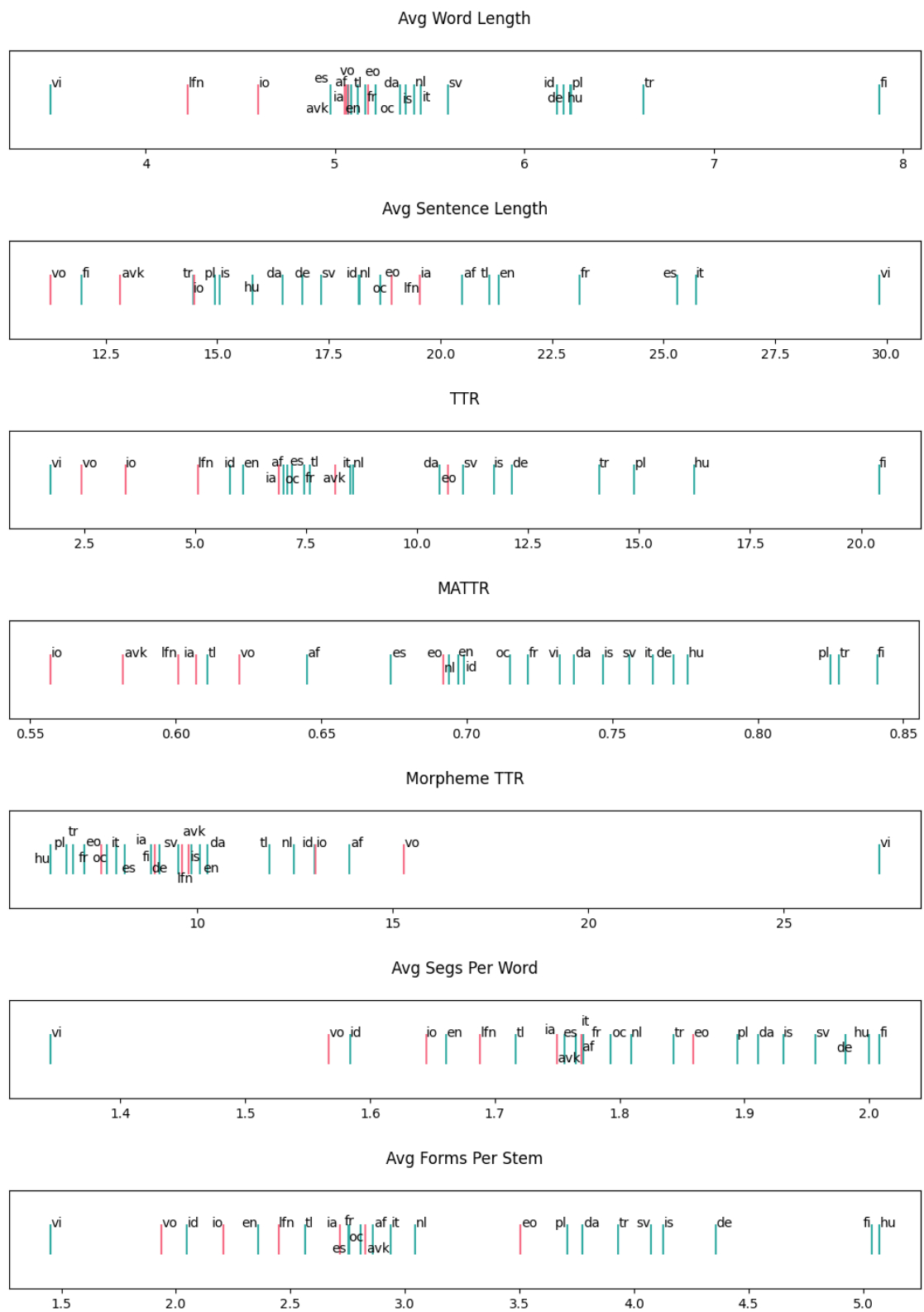
## 4.1 Results of Feature Engineering

Table 4.1 shows the full set of features for each language of the dataset and their respective measurements rounded to three decimals places. Additionally, the column *Type* shows their labels as either constructed or natural languages (*con* or *nat*, respectively).

| Corpus | Type | Avg Word Length | Avg Sentence Length | TTR | MATTR | Morpheme TTR | Avg Segs Per Word | Avg Forms Per Lemma | Char Dist Entr | Word Dist Entr | Text Entr | Lex Entr | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Esperanto | con | 5.175 | 18.909 | 10.708 | 0.692 | 7.535 | 1.859 | 3.506 | 4.164 | 10.923 | 4.435 | 6.303 | 7.0 |
| Dutch | nat | 5.419 | 18.194 | 8.559 | 0.694 | 12.467 | 1.809 | 3.044 | 4.117 | 10.593 | 4.389 | 6.407 | 6.7 |
| Icelandic | nat | 5.375 | 15.055 | 11.727 | 0.747 | 9.847 | 1.931 | 4.128 | 4.468 | 11.512 | 5.455 | 5.993 | 6.3 |
| Polish | nat | 6.248 | 14.951 | 14.890 | 0.825 | 6.639 | 1.894 | 3.709 | 4.553 | 12.905 | 5.072 | 5.611 | 5.8 |
| French | nat | 5.160 | 23.120 | 7.461 | 0.721 | 7.109 | 1.771 | 2.759 | 4.179 | 10.711 | 4.104 | 6.256 | 6.7 |
| Volapük | con | 5.072 | 11.266 | 2.455 | 0.622 | 15.287 | 1.567 | 1.938 | 4.256 | 7.666 | 1.281 | 8.702 | 9.0 |
| Afrikaans | nat | 5.067 | 20.496 | 6.987 | 0.645 | 13.894 | 1.770 | 2.861 | 4.072 | 9.993 | 4.757 | 6.639 | 6.9 |
| Vietnamese | nat | 3.498 | 29.835 | 1.749 | 0.732 | 27.439 | 1.344 | 1.453 | 4.855 | 9.717 | 4.768 | 11.878 | 11.4 |
| Occitan | nat | 5.215 | 18.660 | 7.185 | 0.715 | 7.674 | 1.793 | 2.805 | 4.173 | 10.546 | 3.530 | 6.741 | 7.1 |
| English | nat | 5.087 | 21.301 | 6.079 | 0.697 | 10.049 | 1.661 | 2.360 | 4.167 | 10.673 | 4.771 | 7.106 | 7.6 |
| Italian | nat | 5.455 | 25.727 | 8.505 | 0.764 | 7.919 | 1.765 | 2.939 | 4.029 | 11.308 | 4.573 | 5.563 | 6.1 |
| Interlingua | con | 5.050 | 19.547 | 6.880 | 0.607 | 8.904 | 1.750 | 2.715 | 4.032 | 10.005 | 3.886 | 6.406 | 6.9 |
| Swedish | nat | 5.597 | 17.322 | 11.031 | 0.756 | 9.496 | 1.957 | 4.074 | 4.294 | 11.488 | 4.830 | 6.207 | 6.5 |
| Lingua Franca Nova | con | 4.221 | 19.532 | 5.063 | 0.601 | 9.769 | 1.688 | 2.448 | 3.912 | 9.316 | 4.471 | 7.716 | 8.4 |
| Danish | nat | 5.346 | 16.466 | 10.517 | 0.737 | 10.259 | 1.911 | 3.776 | 4.197 | 11.274 | 5.037 | 6.408 | 6.7 |
| Hungarian | nat | 6.242 | 15.782 | 16.234 | 0.776 | 6.240 | 2.000 | 5.071 | 4.543 | 12.443 | 5.208 | 5.584 | 5.9 |
| Indonesian | nat | 6.173 | 18.164 | 5.782 | 0.699 | 12.993 | 1.584 | 2.048 | 4.072 | 11.142 | 3.982 | 7.285 | 7.3 |
| Tagalog | nat | 5.119 | 21.102 | 7.593 | 0.611 | 11.830 | 1.717 | 2.563 | 3.895 | 9.991 | 4.374 | 6.803 | 7.0 |
| Turkish | nat | 6.630 | 14.458 | 14.097 | 0.828 | 6.805 | 1.843 | 3.932 | 4.386 | 13.151 | 4.729 | 5.028 | 5.4 |
| Ido | con | 4.594 | 14.484 | 3.433 | 0.557 | 13.024 | 1.645 | 2.208 | 4.077 | 8.055 | 1.266 | 7.471 | 8.2 |
| Spanish | nat | 4.978 | 25.315 | 7.085 | 0.674 | 8.131 | 1.756 | 2.752 | 4.046 | 10.327 | 4.066 | 6.059 | 6.6 |
| Finnish | nat | 7.874 | 11.969 | 20.409 | 0.841 | 8.806 | 2.008 | 5.037 | 4.144 | 13.729 | 4.529 | 4.994 | 5.4 |
| German | nat | 6.206 | 16.907 | 12.128 | 0.771 | 9.026 | 1.981 | 4.356 | 4.230 | 11.601 | 4.530 | 5.323 | 5.7 |
| Kotava | con | 5.060 | 12.824 | 8.153 | 0.582 | 9.611 | 1.769 | 2.827 | 4.186 | 10.287 | 3.759 | 7.809 | 8.0 |

Table 4.1: Feature set

To visualize these features individually,

## Avg Word Length



## Avg Sentence Length



## TTR



## MATTR



## Morpheme TTR



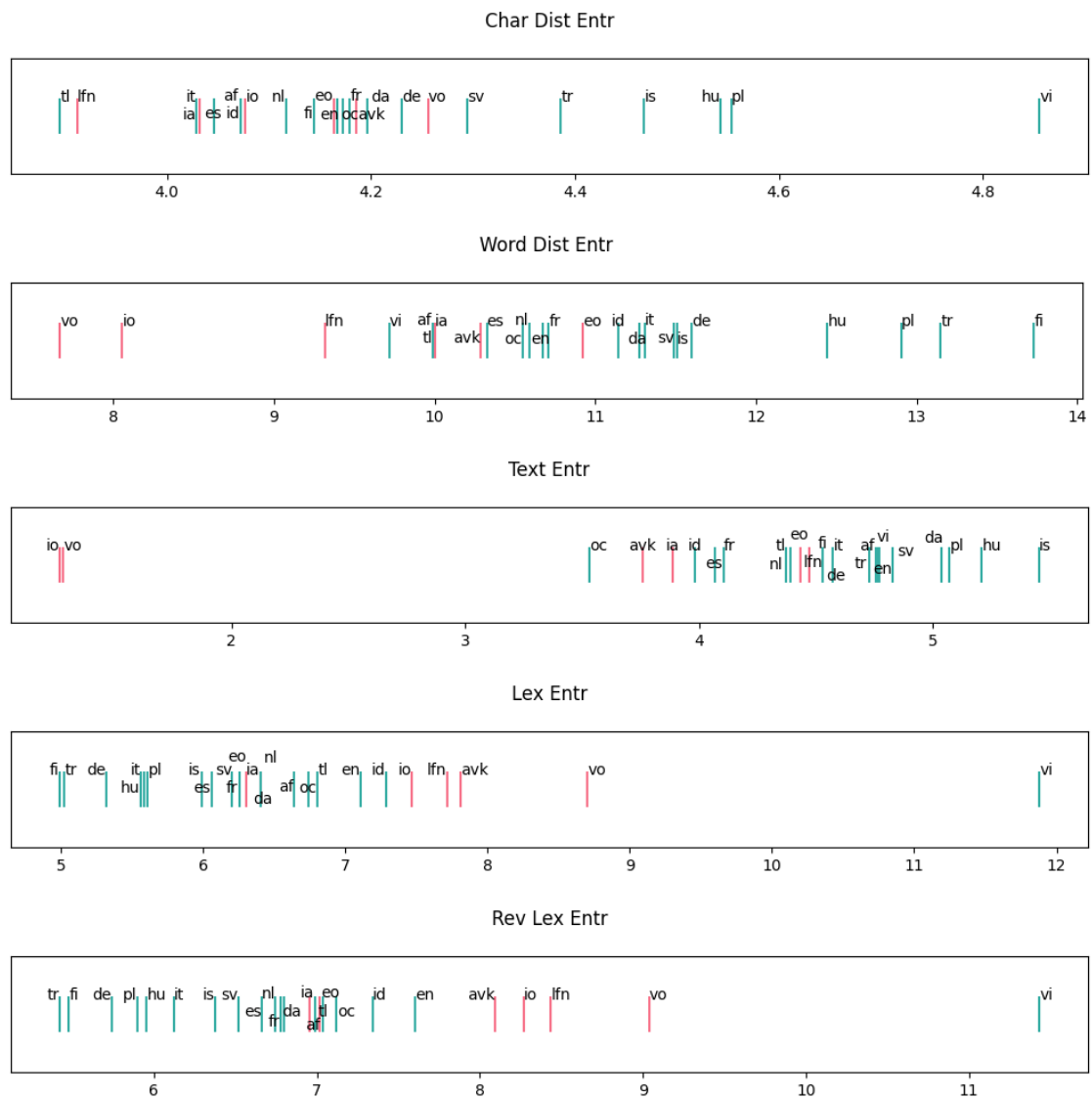## Avg Segs Per Word



## Avg Forms Per Stem

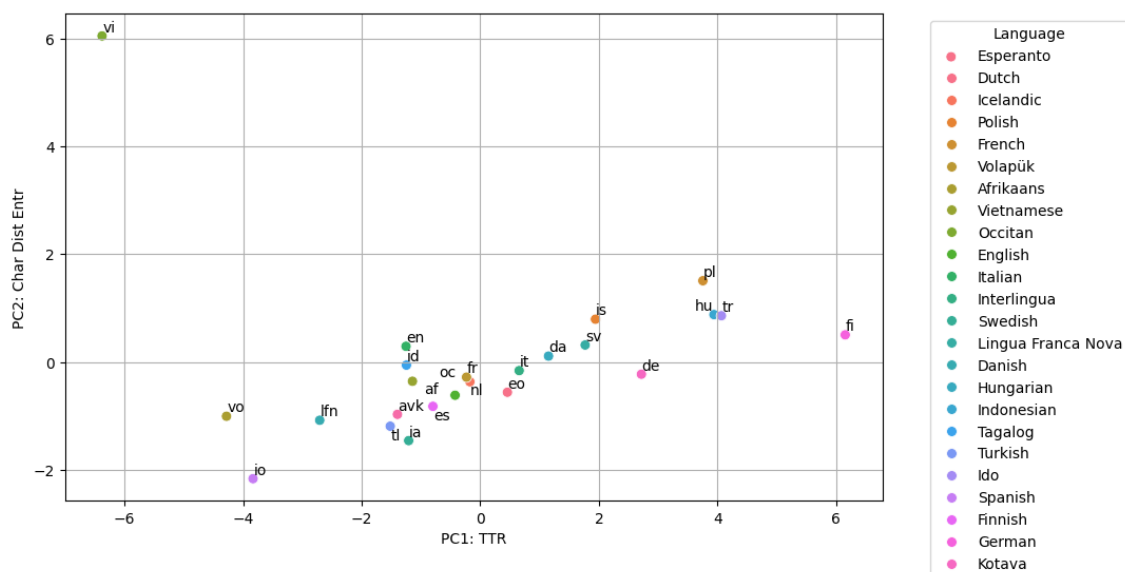Figure 4.1: 1-Dimensional Plots for Feature Distributions

Figure 4.2: Principal Component Analysis on the feature set.

## 4.2   Results of Dimensionality Reduction

| | Avg Word Length | Avg Sentence Length | TTR | MATTR | MTLD | Morpheme TTR | Avg Segs Per Word | Avg Form |
|------|------|------|------|------|------|------|------|------|
| PC1 | 0.312 | -0.152 | 0.343 | 0.271 | 0.284 | -0.25 | 0.318 | 0.328 |
| PC2 | -0.048 | 0.243 | 0.062 | 0.364 | 0.239 | 0.365 | -0.154 | 0.017 |

Table 4.2: Loadings for Each Principal Component

A script was used to increase readability of the text in the graph[4].

## 4.3   Results of Classification

| | $F_1$-score | Precision | Recall |
|------|------|------|------|
| Decision Tree | | | |
| Random Forest | | | |

Table 4.3: Evaluation Results for Supervised Binary Classifiers

## 4.4   Results of Anomaly Detection

| | $F_1$-score | Precision | Recall |
|------|------|------|------|
| Isolation Forest | | | |
| One-Class SVM | | | |
| Local Outlier Factor | | | |

Table 4.4: Evaluation Results for Unsupervised Anomaly Detection Models

---

[4]https://github.com/Phlya/adjustText
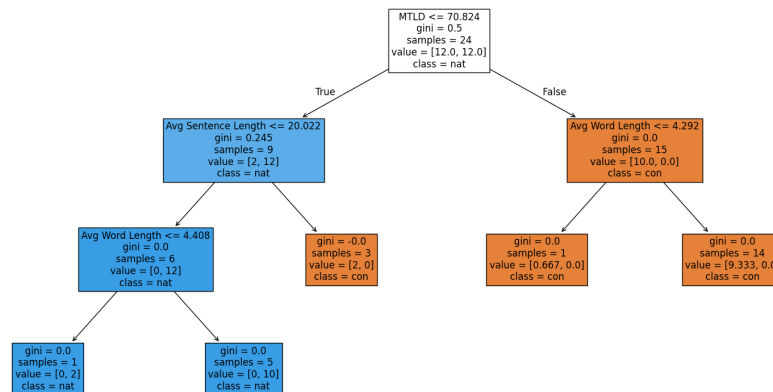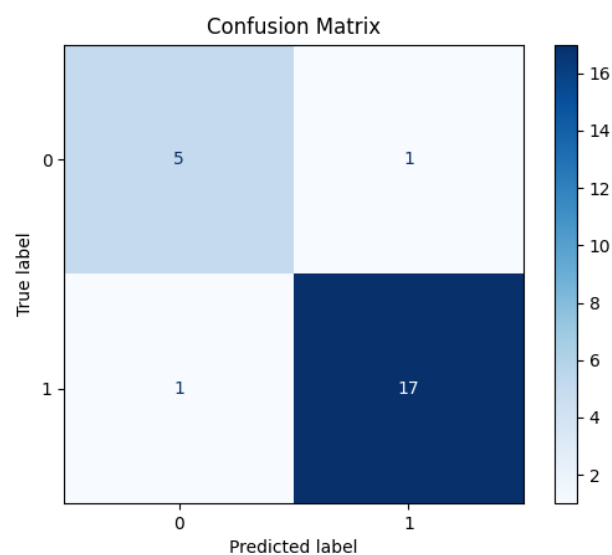
Figure 4.3: Decision Tree Classifier



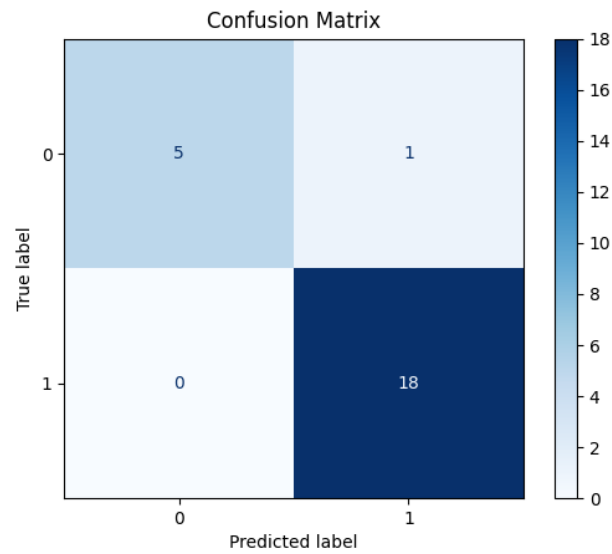Figure 4.4: Confusion Matrix for Decision Tree Classifier

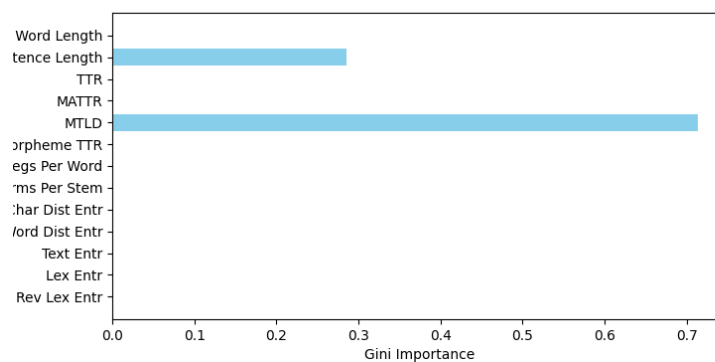Figure 4.5: Confusion Matrix for Random Forest Classifier



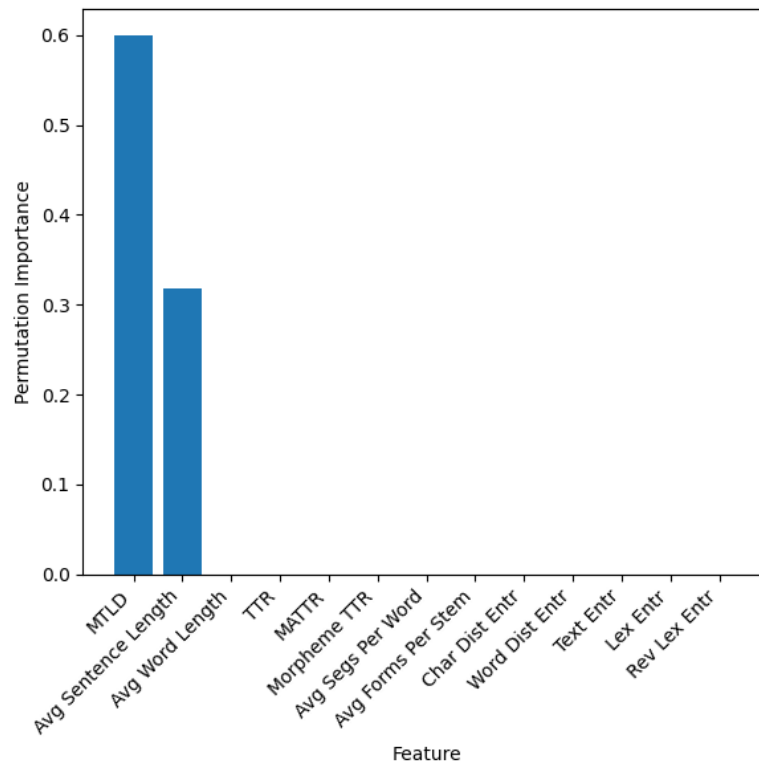Figure 4.6: Gini Importances for Random Forest

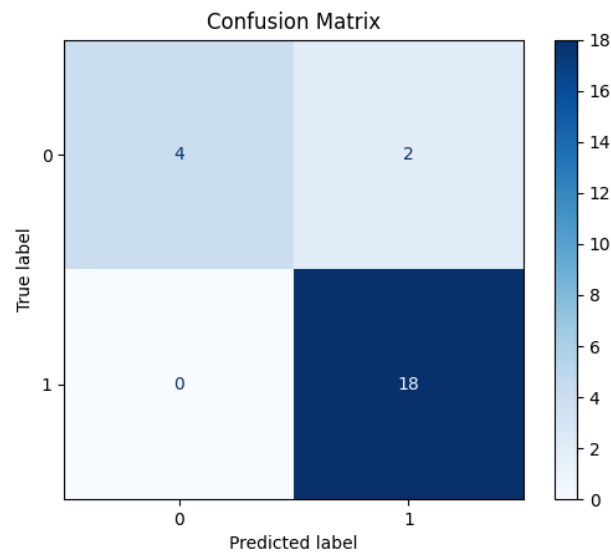Figure 4.7: Permutation Importances for Random Forest



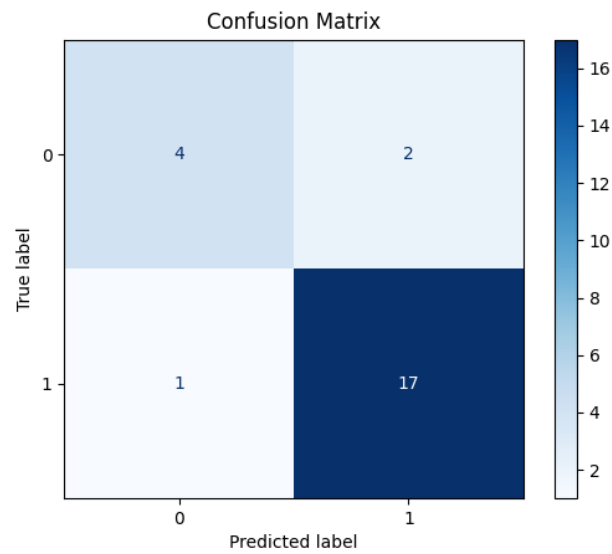Figure 4.8: Confusion Matrix for One-Class SVM

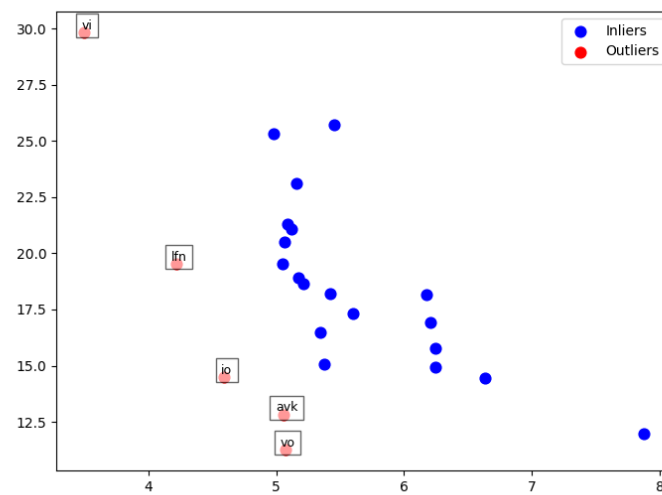Figure 4.9: Confusion Matrix for Isolation Forest
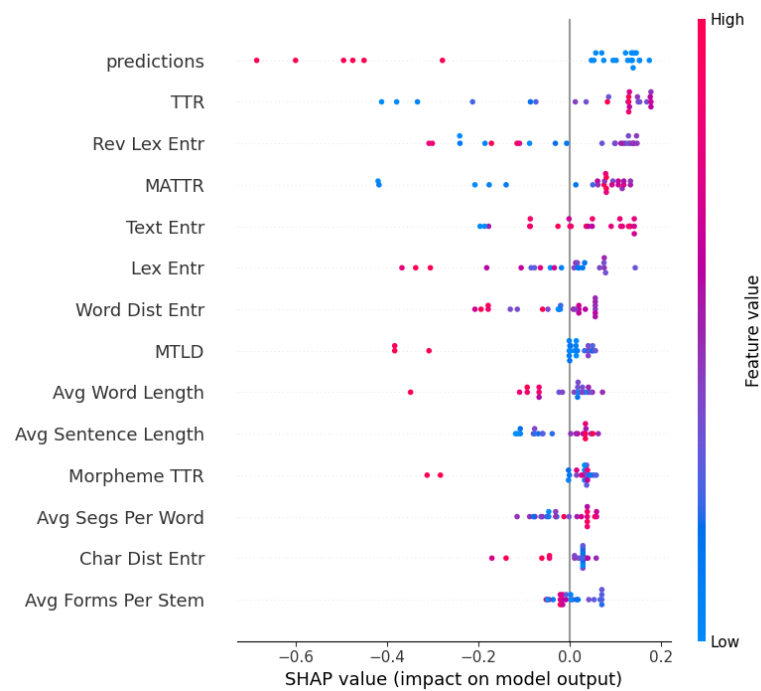


Figure 4.10: Anomalies Detected by Isolation Forest
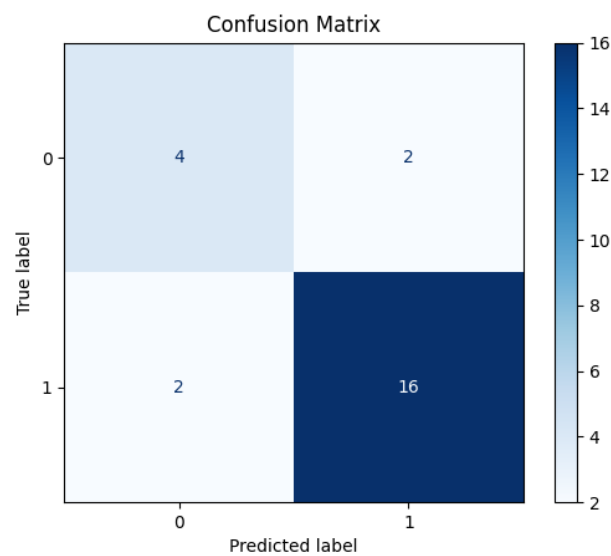
Figure 4.11: SHAP Values of Isolation Forest



Figure 4.12: Confusion Matrix for Local Outlier Factor

## 4.5    Feature Importances

# 5   Discussion

# 6   Conclusion

## 6.1   Future Work

The research presented in this thesis is far from encompassing all there is to the topic of defining language, and distinguishing between constructed and natural language. At present, this is an area of research with ample room for potential development.

Limiting factors: number of languages and which languages/language families, lack of real parallel corpora, problems associated with low-resource languages, relatively narrow scope of experimentation,

# 7 Acknowledgments

I would like to thank ....

# References

Adelman, Michael J. (2014). "Constructed Languages and Copyright: A Brief History and Proposal for Divorce". In: *Harvard Journal of Law & Technology* 27, p. 543. URL: https://api.semanticscholar.org/CorpusID:58553165.

Attardi, Giuseppe (2015). *WikiExtractor*. https://github.com/attardi/wikiextractor.

Ball, Douglas (2015). "Constructed languages". In: *The Routledge Handbook of Language and Creativity*. Ed. by Rodney H Jones. Routledge. Chap. 8. DOI: 10.4324/9781315694566.ch8.

Bausani, A. (1974). *Le lingue inventate: Linguaggi artificiali, linguaggi segreti, linguaggi universali*. Collana di studi umanistici 'Ulisse'. Ubaldini. ISBN: 9788834003879. URL: https://books.google.de/books?id=z4GAngEACAAJ.

Bestgen, Yves (2024). "Measuring Lexical Diversity in Texts: The Twofold Length Problem". In: *Language Learning* 74.3, pp. 638–671. DOI: https://doi.org/10.1111/lang.12630. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/lang.12630. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/lang.12630.

BLANKE, DETLEV (1989). "Planned languages – a survey of some of the main problems". In: *Aspects of the Science of Planned Languages*. Ed. by Klaus Schubert. Berlin, New York: De Gruyter Mouton, pp. 63–88. ISBN: 9783110886115. DOI: doi:10.1515/9783110886115.63. URL: https://doi.org/10.1515/9783110886115.63.

Boiar, Daniel, Thomas Liebig, and Erich Schubert (2022). *LOSDD: Leave-Out Support Vector Data Description for Outlier Detection*. arXiv: 2212.13626 [cs.LG]. URL: https://arxiv.org/abs/2212.13626.

Chomsky, Noam (1957). *Syntactic Structures*. Berlin, Boston: De Gruyter Mouton. ISBN: 9783112316009. DOI: doi:10.1515/9783112316009. URL: https://doi.org/10.1515/9783112316009.

Christiansen, M.H., C. Collins, and S. Edelman (2009). *Language Universals*. Oxford University Press. ISBN: 9780190294113. URL: https://books.google.de/books?id=bCLiBwAAQBAJ.

Cook, V.J. and M. Newson (2007). *Chomsky's Universal Grammar: An Introduction*. Wiley. ISBN: 9781405111874. URL: https://books.google.de/books?id=mguunu3sI-YC.

Covington, Michael and Joe McFall (May 2010). "Cutting the Gordian knot: The moving-average type-token ratio (MATTR)". In: *Journal of Quantitative Linguistics* 17, pp. 94–100. DOI: 10.1080/09296171003643098.

Deng, Yingzhuo et al. (2024). *Robust Principal Component Analysis via Discriminant Sample Weight Learning*. arXiv: 2408.12366 [cs.LG]. URL: https://arxiv.org/abs/2408.12366.

Dhandapani, Aarthi et al. (June 2023). "Lyrics Generation Using LSTM and RNN". In: pp. 371–388. ISBN: 978-981-99-1050-2. DOI: 10.1007/978-981-99-1051-9_24.

Fergadiotis, Gerasimos, Heather Wright, and Thomas West (May 2013). "Measuring Lexical Diversity in Narrative Discourse of People With Aphasia". In: *American journal of speech-language pathology / American Speech-Language-Hearing Association* 22, S397–408. DOI: 10.1044/1058-0360(2013/12-0083).

Fetcey, S. and le Comité Linguistique Kotava (2013). *Kotava: grammaire officielle complète (version III.14)*. URL: ttp://www.kotava.org/fr/fr_pulviropa_000.pdf.

Gobbo, Federico (Jan. 2008). "Planned languages and language planning: The contribution of interlinguistics to cross-cultural communication". In: *Multilingualism and Applied Comparative Linguistics* 2.

– (Sept. 2011). "The Case of Correlatives: A Comparison between Natural and Planned Languages". In: *Journal of Universal Language* 12, p. 34. DOI: 10.22425/jul.2011.12.2.45.

– (Oct. 2016). "Are planned languages less complex than natural languages?" In: *Language Sciences* 60. DOI: 10.1016/j.langsci.2016.10.003.

Goodall, Grant (Sept. 2022). "Constructed Languages". In: *Annual Review of Linguistics* 9. DOI: 10.1146/annurev-linguistics-030421-064707.

Greenberg, Joseph H. (1970). "Language Universals". In: *Theoretical Foundations*. Berlin, Boston: De Gruyter Mouton, pp. 61–112. ISBN: 9783110814644. DOI: doi:10.1515/9783110814644-003. URL: https://doi.org/10.1515/9783110814644-003.

Han, Sunwoo, Brian Williamson, and Youyi Fong (Nov. 2021). "Improving random forest predictions in small datasets from two-phase sampling designs". In: *BMC Medical Informatics and Decision Making* 21. DOI: 10.1186/s12911-021-01688-3.

Jeffrey Punske (editor) Nathan Sanders (editor), Amy V. Fountain (editor) (2020). *Language Invention in Linguistics Pedagogy*. Oxford Uni-

versity Press. ISBN: 0198829876,9780198829874. URL: `http://gen.lib.rus.ec/book/index.php?md5=0B5CF2BFC00DCB569EBA10BD96AD68D4`.

Large, J.A. (1985). *The Artificial Language Movement*. Language library. B. Blackwell. ISBN: 9780631144977. URL: `https://books.google.de/books?id=xaeCQgAACAAJ`.

Libert, Alan (Jan. 2016). "On Pragmemes in Artificial Languages". In: pp. 375–389. ISBN: 978-3-319-43490-2. DOI: `10.1007/978-3-319-43491-9_20`.

Liu, Fei Tony, Kai Ting, and Zhi-Hua Zhou (Jan. 2009). "Isolation Forest". In: pp. 413–422. DOI: `10.1109/ICDM.2008.17`.

Mairal, Ricardo and Juana Gil (Jan. 2006). *Linguistic Universals*. ISBN: 9780521837095. DOI: `10.1017/CBO9780511618215`.

McCarthy, Philip and Scott Jarvis (May 2010). "MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment". In: *Behavior research methods* 42, pp. 381–92. DOI: `10.3758/BRM.42.2.381`.

Novikov, Philipp (July 2022). "Constructed Languages as Semantic and Semiotic Systems". In: *RUDN Journal of Language Studies, Semiotics and Semantics* 13, pp. 455–467. DOI: `10.22363/2313-2299-2022-13-2-455-467`.

Okrent, Arika (2009). *In the Land of Invented Languages: Esperanto Rock Stars, Klingon Poets, Loglan Lovers, and the Mad Dreamers who Tried to Build a Perfect Language*. Spiegel & Grau. ISBN: 9780385527880. URL: `https://books.google.de/books?id=E3UE9IoW27AC`.

Oktafiani, Rian, Arief Hermawan, and Donny Avianto (Feb. 2024). "Max Depth Impact on Heart Disease Classification: Decision Tree and Random Forest". In: *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)* 8, pp. 160–168. DOI: `10.29207/resti.v8i1.5574`.

Pawlas, Elżbieta and Michał B. Paradowski (Jan. 2020). "Misunderstandings in communicating in English as a lingua franca: Causes, prevention, and remediation strategies". In: pp. 101–122. ISBN: 978-83-66666-28-3. DOI: `10.48226/978-83-66666-28-3`.

Rosenhahn, Bodo and Christoph Hirche (2024). *Quantum Normalizing Flows for Anomaly Detection*. arXiv: `2402.02866 [quant-ph]`. URL: `https://arxiv.org/abs/2402.02866`.

Salman, Ahmed and Waleed Al-Jawher (Sept. 2024). "Performance Comparison of Support Vector Machines, AdaBoost, and Random Forest for Sentiment Text Analysis and Classification". In: *Journal Port Science Research* 7, pp. 300–311. DOI: `10.36371/port.2024.3.8`.

Sanders, Nathan (Sept. 2016). "Constructed languages in the classroom". In: *Language* 92, e192–e204. DOI: 10.1353/lan.2016.0055.

Schreyer, Christine and David Adger (Mar. 2021). "Comparing prehistoric constructed languages: World-building and its role in understanding prehistoric languages". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 376. DOI: 10.1098/rstb.2020.0201.

Schubert, Klaus (Jan. 1989). "Interlinguistics – Its Aims, Its Achievements, and Its Place in Language Science". In: pp. 7–44. ISBN: 9783110886115. DOI: 10.1515/9783110886115.7.

Schubert, Klaus et al. (Jan. 2001). *Planned Languages: From Concept to Reality.*

Shannon, C. E. (1949). *A Mathematical Theory of Communication.* Vol. 27, pp. 379–423.

Smit, Peter et al. (Jan. 2014). "Morfessor 2.0: Toolkit for statistical morphological segmentation". In: pp. 21–24. DOI: 10.3115/v1/E14-2006.

Tonkin, Humphrey (Apr. 2015). "Language Planning and Planned Languages: How Can Planned Languages Inform Language Planning?" In: *Interdisciplinary Description of Complex Systems* 13, pp. 193–199. DOI: 10.7906/indecs.13.2.1.

Virpioja, Sami et al. (2013). "Morfessor 2.0: Python Implementation and Extensions for Morfessor Baseline". In: URL: https://api.semanticscholar.org/CorpusID:36074039.

Wilkins, John S. (1968). "An essay towards a real character, and a philosophical language, 1668". In: URL: https://api.semanticscholar.org/CorpusID:161991811.

Xu, Hongzuo et al. (Dec. 2023). "Deep Isolation Forest for Anomaly Detection". In: *IEEE Transactions on Knowledge and Data Engineering* 35.12, pp. 12591–12604. ISSN: 2326-3865. DOI: 10.1109/tkde.2023.3270293. URL: http://dx.doi.org/10.1109/TKDE.2023.3270293.

# 8 Appendices

Here I...

| Language | Number of Words | Number of sentences |
| --- | --- | --- |
| Icelandic | 629995 | 41847 |
| German | 629987 | 37261 |
| Polish | 629997 | 42138 |
| Ido | 629990 | 43496 |
| Afrikaans | 629994 | 30737 |
| Kotava | 617400 | 48145 |
| Hungarian | 629946 | 39916 |
| Lingua Franca Nova | 628683 | 32188 |
| Danish | 629999 | 38260 |
| Spanish | 629978 | 24886 |
| Interlingua | 629996 | 32229 |
| French | 629983 | 27248 |
| Occitan | 629998 | 33762 |
| Esperanto | 629994 | 33317 |
| Dutch | 629997 | 34627 |
| Turkish | 629995 | 43573 |
| English | 629958 | 29574 |
| Tagalog | 629989 | 29855 |
| Swedish | 629998 | 36370 |
| Vietnamese | 629958 | 21115 |
| Italian | 629987 | 24487 |
| Volapük | 629999 | 55920 |
| Indonesian | 629997 | 34683 |
| Finnish | 629994 | 52637 |

Table 8.1: Lengths of each language's text after pre-processing.